

An Immersed Boundary Fourier Pseudo-spectral Method for Simulation of Confined Two-dimensional Incompressible Flows

Fereidoun Sabetghadam^{*}

Mehdi Badri, Shervin Sharafatmandjor, Hosnieh Kor

Mechanical and Aerospace Engineering Department, Science and Research Branch, Azad University (IAU), Tehran, Iran.

Abstract

The present paper is devoted to implementation of the immersed boundary technique into the Fourier pseudo-spectral solution of the vorticity-velocity formulation of the two-dimensional incompressible Navier–Stokes equations. The immersed boundary conditions are implemented via direct modification of the convection and diffusion terms, and therefore, in contrast to many other similar methods, there is not an explicit external forcing function in the present formulation. The desired immersed boundary conditions are approximated on some regular grid points, using different orders (up to second-order) polynomial extrapolations. At the beginning of each timestep, the solenoidal velocities (also satisfying the desired immersed boundary conditions), are obtained and fed into a conventional pseudo-spectral solver, together with a modified vorticity. The zero-mean pseudo-spectral solution is employed, and therefore, the method is applicable to the confined flows with zero mean velocity and vorticity, and without mean vorticity dynamics. In comparison to the classical Fourier pseudo-spectral solution, the method needs $\mathcal{O}(4(1 + \log N)N)$ more operations for boundary condition settings. Therefore, the computational cost of the method, as a whole, is scaled by $(N \log N)$. The classical explicit fourth-order Runge–Kutta method is used for time integration, and the boundary conditions are set at the beginning of each sub-step, in order to increasing the time accuracy. The method is applied to some fixed and moving boundary problems, with different orders of boundary conditions; and in this way, the accuracy and performance of the method are investigated and compared with the classical Fourier pseudo-spectral solutions.

Key words: Fourier pseudo-spectral solution; Two-dimensional Navier–Stokes equations; Vorticity–velocity formulation; Immersed boundary method; Solenoidal velocities; Moving boundary problems

1 Introduction

Fourier pseudo-spectral solution of the vorticity-based formulations of the Navier–Stokes equations (NSE) have been used widely in the two-dimensional incompressible flow simulations [13]. However, the classical implementations are limited to the regular domains with simple coordinate-coinciding boundaries and periodic boundary conditions. Now, recent advances in the immersed and embedded boundary techniques have raised hopes of extending the range of applicability of these methods to the more general domains and boundary conditions [8,41,31].

With the best knowledge of the authors, the immersed boundary method was applied into the vorticity–stream function formulation of the NSE by Calhoun [11,12] for the first time. In the Calhoun’s work the immersed surfaces are introduced, and the overall mass balance is satisfied, by definition of an appropriate distribution of vorticity source term. More or less similar line was followed in the work of Russell and Wang [40]. They decomposed the effects of a solid wall into a no-slip condition, satisfied by the Thom’s rule, and a no-penetration condition, imposed by a boundary element method (which satisfied the overall mass balance). In the work of Linnick and Fasel [34], a higher-order compact method was used, and a source term was defined in crossing the discontinuities, which was obtained from a jump function. Recently, Wang *et al.* [47] applied the *direct forcing* idea of Mohd-Yusof [37] into the vorticity–velocity formulation of the NSE. They added an explicit vorticity source term to the vorticity transport equation, which was obtained by taking curl of the forcing functions of the momentum equations in the primitive variables form of the NSE. However, all the above methods were based upon the finite-difference or finite-volume spatial discretization.

In the pseudo-spectral solutions, the volume penalization is one of the popular remedies, which has been used several times for implementation of the no-slip condition. The method was first proposed in the primitive variables formulation of the NSE by Arquis and Caltagirone [4], and then re-formulated by Angot [3]. In the next years the method was extended to the vorticity–velocity formulation [29,42,28], and used for the fixed, as well as moving boundary problems [31,43,29,16].

A new immersed boundary method is proposed in the present paper, in which the arbitrary immersed velocity boundary conditions (including the no-slip condition), are introduced into the Fourier pseudo-spectral solution of the vorticity–velocity formulation of the NSE, without explicit addition of external forcing functions. Instead of the conventional forcing functions, the immersed boundaries are implemented by direct modification of the convection and diffusion terms of the vorticity transport equation in such a way that can

*

Email address: fsabet@srbiau.ac.ir (Fereidoun Sabetghadam).

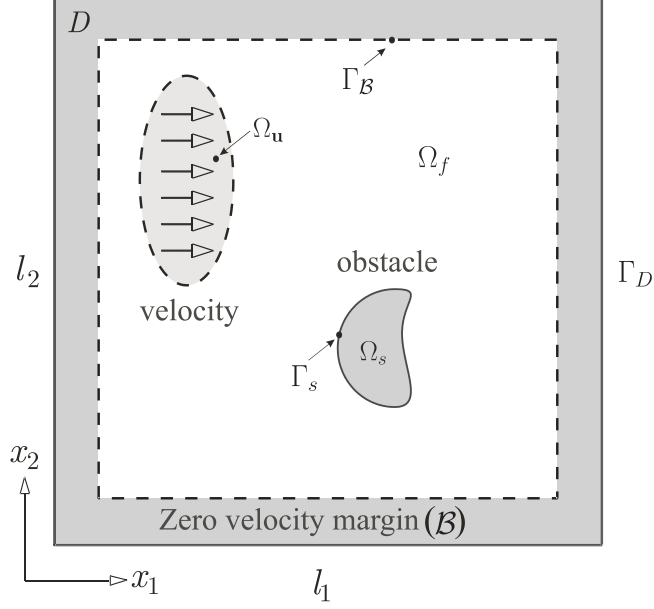


Fig. 1. The flow domain Ω_f , together with fixed or moving obstacle(s) Ω_s , and other given-velocity regions Ω_u , are embedded in the regular solution domain D (with regular boundary Γ_D), via a zero-velocity margin \mathcal{B} .

be implemented to the Fourier pseudo-spectral solutions.

Particularly, it is aimed to use the zero-mean Fourier pseudo-spectral method. Therefore, among many possibilities, our suggested flow configuration is illustrated in Fig. 1. The regular solution domain D contains all the fluid–solid system, including the moving regions of the fluid with given velocity boundary conditions (that is Ω_u), in addition to the fixed or moving immersed solid objects. Moreover, note that the fluid–solid system is embedded in D via a zero-velocity margin \mathcal{B} . This margin is considered to ensure that the flow physics can be followed by the zero-mean Fourier series, namely, to ensure that the velocity and vorticity fields remain zero-mean, and the dynamics of the mean vorticity is zero, during the time integration. In fact, we found that considering such a margin can also be useful in other methods (*e.g.* the finite difference method), in order to overcoming some difficulties related to the appropriate vorticity boundary conditions and finding solenoidal velocities (see the appendix for more discussions about the role of margin \mathcal{B}). In the present work, this margin is obtained by windowing of the velocity fields.

On the other hand, definition of this margin is also in the following of our previous work [41], and the idea of an *infinitely flat* window function, proposed in [8]. However, achieving the spectral rates of convergence is not anticipated for the finite Reynolds numbers (which is the subject of this paper); therefore, it is not needed *infinitely flat* window functions. However, the algebraic rates of convergence of the Fourier series are competing in many practical situations. Although the shape of the inner boundary of \mathcal{B} (*i.e.* Γ_B) is arbitrary, in our

numerical experiments we used rounded rectangular shapes for simplicity. The central core of the method is a conventional pseudo-spectral solver of the two-dimensional incompressible NSE in the vorticity-velocity form. At the beginning of each time step, the desired immersed velocity boundary conditions are imposed via direct modification of the velocity fields. Then the modified vorticity (which will be called the conditioned vorticity), is constructed directly from the modified velocities. The modified vorticity is fed directly into the pseudo-spectral solver, which gives the vorticity field in the next time. Time integration is performed using the classical explicit fourth-order Runge–Kutta method to keep the method as simple as possible, and to avoid some difficulties associated with the implicit formulations. In comparison to the classical pseudo-spectral method, the present method needs $\mathcal{O}(4(1 + \log N)N)$ more operations for boundary condition settings, and therefore, the computational cost of the method, as a whole, is scaled by $(N \log N)$. The paper is continued by presenting the mathematical formulations of the classical Fourier pseudo-spectral method and the suggested algorithm for imposing the immersed velocity boundary conditions. Because of its crucial role, the boundary condition setting process is presented in details, in an individual section. As our numerical experiments, the method is implemented into some fixed as well as moving boundary problems, with the surfaces which are coinciding and non-coinciding with the regular grids. Finally, some basic questions about the validity of the method are addressed in an appendix.

2 Mathematical Formulation

The mathematical and numerical frameworks of the method are described in this section. Beginning from the classical Fourier pseudo-spectral formulation, the suggested modifications for imposing the immersed boundaries, and then embedding the solution domain into the regular domain are explained.

2.1 The Fourier pseudo-spectral formulation

According to Fig. 1, for a two-dimensional velocity vector $\mathbf{u} = (u_1, u_2)$, defined on the regular closure $\bar{D} = (D \cup \Gamma_D)$, the dynamics of the vorticity vector $\omega = (0, 0, \omega_z = \omega \hat{\mathbf{e}}_z = \nabla \times \mathbf{u})$ is obtainable from time integration of the vorticity transport equation

$$\begin{cases} \partial_t \omega + (\mathbf{u} \cdot \nabla) \omega = \nu \nabla^2 \omega & \text{in } D \times (0, T], \\ \omega(\mathbf{x}, t = 0) = \omega_0(\mathbf{x}) & \text{for } \mathbf{x} \in \bar{D}, \end{cases} \quad (1)$$

while the velocity vector \mathbf{u} satisfies the following Poisson's problem with Dirichlet boundary conditions:

$$\begin{cases} \nabla^2 \mathbf{u} = \hat{\mathbf{e}}_z \times \nabla \omega & \text{in } D, \\ \mathbf{u}(\Gamma_D) = \mathbf{u}_{\Gamma_D}. \end{cases} \quad (2)$$

One of the advantages of the above vorticity-velocity formulation, in comparison to *e.g.* many primitive variable formulations, is the possibility of decomposition of the kinematics and dynamics of the flow field at each time instant. In fact, for any arbitrary distribution $\omega \in \mathcal{L}^2(D)$, the physical (divergence-free) velocity vector is obtainable from solution of Eq. (2), if the appropriate boundary conditions are imposed (see the appendix or [20] for more discussions). As it will be seen, this issue has a vital role in construction of the physical immersed velocities in the present method.

On the other hand, to improve the efficiency of the computations, it is convenient to change the vorticity transport equation (1) to

$$\partial_t \omega = \underbrace{\nu \nabla^2 \omega}_{\mathbf{L}} - \underbrace{\frac{\partial^2}{\partial x_1 \partial x_2} (u_2^2 - u_1^2)}_{\mathbf{N}_1} + \underbrace{\left(\frac{\partial^2}{\partial x_2^2} - \frac{\partial^2}{\partial x_1^2} \right) u_1 u_2}_{\mathbf{N}_2}, \quad (3)$$

which in comparison to the classical form (1), saves one fast Fourier transform (FFT) in the pseudo-spectral algorithm. Although this formulation has been used in some other studies, to the best knowledge of the authors, in the pseudo-spectral solution of the incompressible flow, it was proposed by Chasnov [15] for the first time. The diffusion term \mathbf{L} and the non-linear terms, \mathbf{N}_1 and \mathbf{N}_2 , are named for the future references. In fact, the immersed velocity boundary conditions will be introduced to the solution by direct modification of these terms.

For the periodic boundary conditions, the Fourier series provides such an accurate and efficient tool which makes it worthwhile re-formulating the problem in the Fourier space. In this way, the vorticity transport equation (3) recasts

$$\begin{cases} d_t \hat{\omega} = - \underbrace{\nu |\mathbf{k}|^2 \hat{\omega}}_{\hat{\mathbf{L}}} - \underbrace{k_1 k_2 (\widehat{u_1^2 - u_2^2})}_{\hat{\mathbf{N}}_1} + \underbrace{(k_1^2 - k_2^2) \widehat{u_1 u_2}}_{\hat{\mathbf{N}}_2}, \\ \hat{\omega}(\mathbf{k}, t = 0) = \hat{\omega}_0, \end{cases} \quad (4)$$

while the Poisson's problem (2) simplifies to

$$\hat{\mathbf{u}} = -i \frac{\mathbf{k}^\perp}{|\mathbf{k}|^2} \hat{\omega}. \quad (5)$$

In these equations, $(\hat{\cdot})$ stands for the quantities in the Fourier space, $\mathbf{k} = (k_1, k_2)$ is the wavenumber vector with the magnitude $|\mathbf{k}|^2 = k_1^2 + k_2^2$; while $\mathbf{k}^\perp = (-k_2, k_1)$ is the perpendicular wavenumber vector, and $i^2 = -1$. Practically, in the finite dimensional calculations, the nonlinear terms $\hat{\mathbf{N}}_1$ and $\hat{\mathbf{N}}_2$, are constructed (and are de-aliased) in the physical space—the algorithm which is known (and will be referred in this paper) as the pseudo-spectral method. Discretization in time, and time integration of the fully discretized system can be done using an appropriate time marching method. The explicit fourth-order Runge–Kutta method is used in the present work. More or less similar formulations have been used in many efficient and accurate solvers of the periodic flow in regular domains. In the sequel we will modify equations (4) and (5), and suggest an algorithm to use them in the flow configuration of Fig. 1.

2.2 Implementation of the immersed velocity boundary conditions

In the present method, without explicit addition of a forcing function in the right hand side of Eq. (4), the immersed surfaces are introduced by modification of the $\hat{\mathbf{N}}_1$, $\hat{\mathbf{N}}_2$, and $\hat{\mathbf{L}}$. Particularly, it is desired to carry out these modifications such that the velocities remain solenoidal, and in a manner that the method can be implemented easily into a Fourier pseudo-spectral solver. The suggested procedure is summarized in Fig. 2, in which the boundary conditions box (BC) can be more explained by the following remarks:

1. Given the vorticity field $\hat{\omega}^n$, from the initial condition or the last timestep, the velocity vector \mathbf{u} in the regular domain \bar{D} , is obtained from Eq. (5) and two inverse FFTs (box (I)).
2. The velocity vector \mathbf{u} is modified to satisfy all needed immersed velocity boundary conditions (box (II)). This conditioned velocity will be called \mathbf{u}^{BC} .

The modifications are carried out by local extrapolations, extension, and windowing of \mathbf{u} ; and will be explained in details in § 2.3. Note that \mathbf{u}^{BC} is neither necessarily solenoidal, nor its mean value is necessarily zero at this point.

3. The conditioned vorticity ω^{BC} is re-calculated from \mathbf{u}^{BC} (that is, $\omega^{\text{BC}} = \nabla \times \mathbf{u}^{\text{BC}}$), as it is shown in box (III).

There are two main reasons for this step. Firstly, the solenoidal velocities can be obtained from this conditioned vorticity in the next steps, provided that the appropriate boundary conditions are implemented (see the appendix and the discussions in there); and secondly, the vorticity will be needed in the subsequent pseudo-spectral algorithm.

Although ω^{BC} can be found by any method (*e.g.* the finite difference), to preserve the spectral accuracy, and because the vorticity in the Fourier

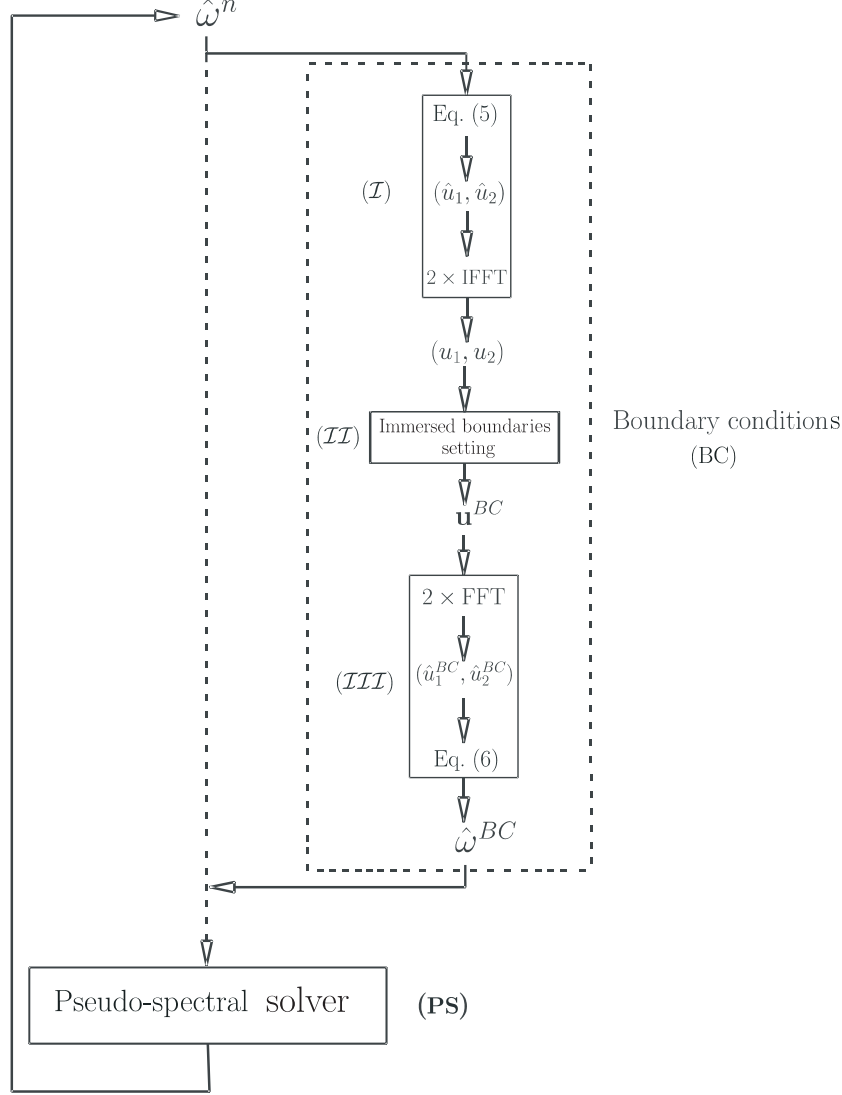


Fig. 2. The main steps of the proposed algorithm. The box **(PS)** contains a classical Fourier pseudo-spectral solver (calculation of the right hand side of Eq. (4), solution of Eq. (5), de-aliasing, time integration...). Therefore, a classical Fourier pseudo-spectral solution with periodic boundary conditions can be retrieved by switching-off the boundary condition setting box, and following the dashed line.

space is needed in the subsequent steps of the pseudo-spectral algorithm, calculation in the Fourier space is suggested here. In this way

$$\hat{\omega}^{BC} = i(k_1 \hat{u}_2^{BC} - k_2 \hat{u}_1^{BC}). \quad (6)$$

Note that it is aimed to simulate the confined flows, and therefore, the vorticity field has zero-mean according to the Stokes theorem; the fact that legitimates use of the above equation. Moreover, note that $\omega^{BC} = \text{IFFT}\{\hat{\omega}^{BC}\}$ is automatically defined on \bar{D} , it is double periodic, and it has zero mean—the properties that makes it *ready to use* for the subsequent Fourier pseudo-

spectral steps.

4. The conditioned vorticity $\hat{\omega}^{\text{BC}}$ is fed into the classical pseudo-spectral procedure (that is, box **(PS)** in Fig. 2), in which the solenoidal velocity vector $\hat{\mathbf{u}}_{\text{Sol}}^{\text{BC}}$ is calculated from

$$\hat{\mathbf{u}}_{\text{Sol}}^{\text{BC}} = -i \frac{\mathbf{k}^\perp}{|\mathbf{k}|^2} \hat{\omega}^{\text{BC}}, \quad (7)$$

and these velocities, in addition to the conditioned vorticity $\hat{\omega}^{\text{BC}}$ are substituted into the modified vorticity transport equation

$$\begin{cases} d_t \hat{\omega} = -\nu |\mathbf{k}|^2 \hat{\omega}^{\text{BC}} - k_1 k_2 (\widehat{u_1^2 - u_2^2})_{\text{Sol}}^{\text{BC}} + (k_1^2 - k_2^2) (\widehat{u_1 u_2})_{\text{Sol}}^{\text{BC}}, \\ \hat{\omega}(\mathbf{k}, t = 0) = \hat{\omega}^{\text{BC}}. \end{cases} \quad (8)$$

Time integration of the above equation yields the new vorticity field $\hat{\omega}^{n+1}$ which closes the algorithm loop.

An attractive feature of the above procedure is that the boundary conditions box can be added easily to a classical pseudo-spectral solver without any change in the internal structure. In fact, the box **(PS)** contains all steps of a classical pseudo-spectral solution (as it was formulated in §2.1). Moreover, note that in addition to the solid obstacles, the given immersed fluid velocities can be implemented as well.

On the other hand, as it was mentioned earlier, implementation of the boundary conditions (that is, box **(II)**), can imposes some discontinuities to the velocity field. Although the next steps will remove these discontinuities (as it will be proven in the appendix), the boundary conditions will change a bit. Similar problem was observed in many other immersed boundary methods, and caused emergence of some methods like the multi-direct forcing method [47,35], or the implicit forcing method [32,45] in the last years. In the present work, we repeat the above algorithm \mathcal{N}_r times to overcome this difficulty; where $1 \leq \mathcal{N}_r \leq 3$ depending on the flow. By \mathcal{N}_r times repetition of the boundary condition settings (and also by development of the solution), the velocity field converges to a solenoidal field which satisfies (approximately) the desired velocity boundary conditions.

In our numerical experiments that are presented in this paper, the fourth-order Runge–Kutta method is used for time integration, and in order to increasing the time accuracy, the boundary conditions are set at the beginning of each Runge–Kutta sub-step.

2.3 Boundary conditions setting

This section is devoted to a full description of the method used in setting of the immersed velocity boundary conditions (that is, modifying \mathbf{u} into \mathbf{u}^{BC} ,

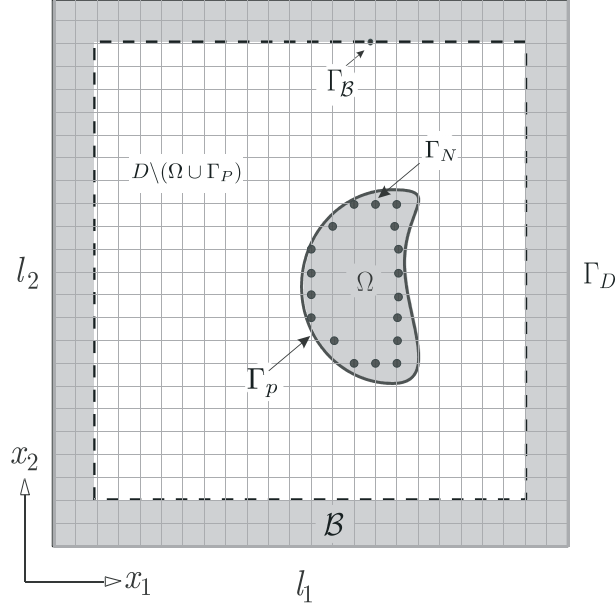


Fig. 3. The numerical boundary Γ_N corresponding to the physical boundary Γ_P . The immersed boundary conditions \mathbf{u}_{pb} are given on Γ_P .

mentioned in item 2 of section 2.2). The method includes a local extrapolation, borrowed from the finite difference-based immersed boundary methods [44], followed by an extension and windowing process, which we used in our previous work [41]. Particularly, this combination is chosen in order to achieving different (desired) orders of *local accuracies* in boundary condition settings, and the needed flexibility in treating the moving boundary problems. In this way, the process of boundary condition setting is divided into the following sub-steps:

- (1) Identification of numerical boundary points.
- (2) Evaluation of the numerical boundary conditions.
- (3) Embedding in the regular domain D .

The details of the above sub-steps are in order. In what follows, we will consider one immersed body. For the multi-object problems, the method can be applied exactly in the same way. Moreover, according to Fig. 3, we assume that the physical velocity boundary conditions \mathbf{u}_{pb} are given on the physical boundary Γ_P , and the solution is sought in $D \setminus (\Omega \cup \mathcal{B})$, and the regular domain D is overlaid by a uniform Cartesian grid (x_i, y_j) .

2.3.1 Identification of numerical boundary points

In the present method, all calculations are performed on a fixed Cartesian grid. Therefore, in the following of our previous work [41], we define the numerical

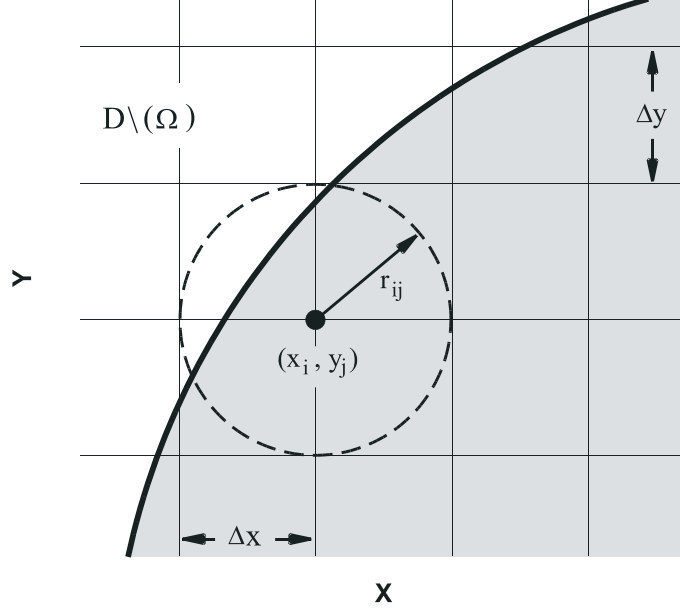


Fig. 4. A numerical boundary point defined on a uniform grid.

boundary points, which play the role of Eulerian points in some fluid–solid interaction methods, which use both the Eulerian and Lagrangian points (see *e.g.* [46]).

Definition 1. A numerical boundary point corresponding to the given physical boundary Γ_P , is a point (x_i, y_j) in the Cartesian grid, if and only if

- i) $(x_i, y_j) \in (\bar{\Omega} = \Omega \cup \Gamma_P)$
- ii) \mathcal{C}_{ij} contains at least one point from $D \setminus \bar{\Omega}$

where \mathcal{C}_{ij} is a circle of radius $r_{ij} = \min(\Delta x, \Delta y)$, centered in (x_i, y_j) . The definition is illustrated in Fig. 4 (for a uniform grid), and for some more details one can see [41]. The set of all numerical boundary points will be called the numerical boundary Γ_N , and that part of the Cartesian grid which is surrounded by Γ_N will be called the numerical immersed domain Ω_N .

For the fixed boundary problems, it is just needed to determine the numerical boundaries once for all computations, while for the moving boundary problems they should be updated in the beginning of each timestep with the computational cost of $\mathcal{O}(\mathcal{N}_{\bar{\Omega}})$, where $\mathcal{N}_{\bar{\Omega}}$ is the number of grid points in a box, contains the immersed domain $\bar{\Omega} = \Omega_N \cup \Gamma_N$.

2.3.2 Approximating the boundary conditions

Given the numerical boundary points Γ_N , the next step is to evaluating velocities on these points (will be called \mathbf{u}_{Γ_N}), such that the physical boundary

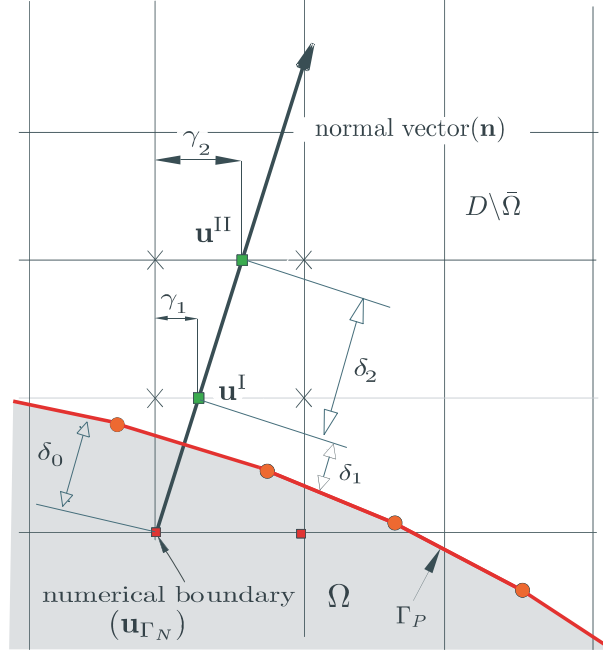


Fig. 5. The geometric parameters, and the interpolation points for a second-order boundary condition setting (*i.e.*, $\mathcal{N}_p = 2$). The auxiliary quantities \mathbf{u}^I and \mathbf{u}^II are obtained from linear interpolations between the neighboring points, which are shown by \times .

conditions \mathbf{u}_{pb} be satisfied approximately. Among several possibilities, an extrapolation method is borrowed from the finite difference-based immersed boundary methods [44], because of its flexibility, and its consistency with our next steps.

Basically, the method is an order \mathcal{N}_p polynomial extrapolation, along the local normal direction \mathbf{n} . According to Fig. 5, given the velocity vector $\mathbf{u} = (u_1, u_2)$, the auxiliary velocities $\mathbf{u}^I, \mathbf{u}^II \dots$ (number of them depends on \mathcal{N}_p), are determined from some linear interpolations between the neighboring points on the Cartesian grid. Then the desired approximation is obtained from

$$\mathbf{u}_{\Gamma_N} = \mathbf{f}_{\mathcal{N}_p}(\mathbf{u}^I, \mathbf{u}^II, \dots; \mathbf{u}_{pb}), \quad (9)$$

where $\mathbf{f}_{\mathcal{N}_p}$ is the polynomial extrapolation function of order \mathcal{N}_p , and \mathbf{u}_{pb} is the given physical velocity boundary condition. The neighboring points which are involved in calculation of the auxiliary velocities are depended on the slope of the normal direction \mathbf{n} (see [44]). In the present paper $\mathcal{N}_p = 0, 1, 2$ are used in our different test cases, and also in different regions of each test case, depending on the desired local accuracies.

As an example, for a second order extrapolation (that is, $\mathcal{N}_p = 2$), according

to Fig. 5, the numerical velocity boundary conditions can be obtained from

$$\mathbf{u}_{\Gamma_N} = \frac{\delta_2(\delta_2 + \delta_3)}{(\delta_1 + \delta_2)P} \mathbf{u}_{\text{pb}} + \frac{\delta_1(\delta_2 + \delta_3)}{\delta_3(\delta_1 + \delta_2)} \mathbf{u}^I - \frac{\delta_1\delta_2}{\delta_3S} \mathbf{u}^{II}, \quad (10)$$

where $S = \delta_1 + \delta_2 + \delta_3$; and \mathbf{u}^I , and \mathbf{u}^{II} are found from linear interpolations. With regard to these extrapolations, the following practical points are worth mentioning:

- (1) Performing these extrapolations showed to be not so easy for complex geometries and moving boundaries. Moreover, for $\mathcal{N}_p > 0$ it may increase the sharpness of the velocity fields, which particularly for the spectral solutions may increase the Gibbs oscillations by triggering the higher wavenumber modes.
- (2) For $\mathcal{N}_p = 0$, it is not any extrapolation indeed, and the boundary condition setting reduces to using a mask function, which substantially simplifies the process; and in many cases, reduces the Gibbs oscillations. However, for the boundaries which are not coinciding with the Cartesian grid, it means zero-order implementation of the boundary conditions, which reduces the global accuracy of the solution, as it will be seen in § 3.

By substitution of \mathbf{u}_{Γ_N} into \mathbf{u} , an auxiliary modified velocity vector \mathbf{u}_{nb} obtains, which is identical to \mathbf{u} in $D \setminus \Omega_N$, and satisfies (approximately) the physical boundary conditions on Γ_P .

2.3.3 Embedding in the regular domain

To take advantages of the pseudo-spectral solutions, the auxiliary modified velocity vector \mathbf{u}_{nb} should be embedded in the regular domain \bar{D} ; and in order to minimizing the Gibbs oscillations, the final modified embedded velocities \mathbf{u}^{BC} should be as smooth as possible. Therefore, referring to Fig. 3, the final modified velocity \mathbf{u}^{BC} should satisfy the following conditions

$$\mathbf{u}^{\text{BC}} = \begin{cases} \mathbf{u}_{\text{nb}} & \text{on } \bar{D} \setminus (\bar{\mathcal{B}} \cup \Omega_N), \\ \mathbf{u}_{\text{pb}} & \text{in } \Omega_N, \\ 0 & \text{in } \mathcal{B}. \end{cases} \quad (11)$$

Moreover, smooth transitions from \mathbf{u}_{nb} to \mathbf{u}_{pb} (in Ω_N), and from \mathbf{u}_{nb} to zero (in \mathcal{B}) are desired, in order to minimizing the Gibbs oscillations. In this context, the following two-step procedure is followed for the sake of flexibility:

- (1) **Extension:** Given the extrapolation functions $\mathbf{f}_{\mathcal{N}_p}$ (for different points of Γ_N and $\Gamma_{\mathcal{B}}$), the desired extensions of \mathbf{u}_{nb} (in Ω_N and \mathcal{B}), are eas-

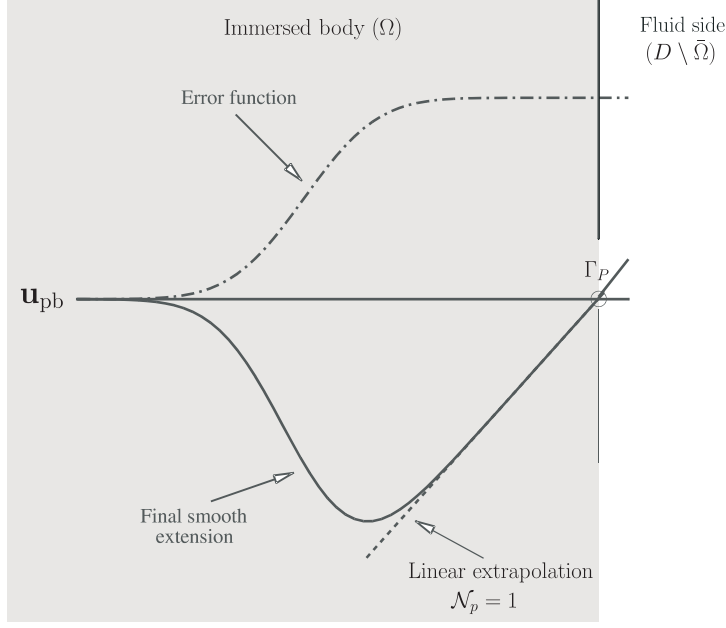


Fig. 6. A sectional view of an extended and periodized field for $\mathcal{N}_p = 1$. The u_i^{BC} are obtained from multiplication of a linear extension (with a slope which is computed from the internal points adjacent to the physical boundary Γ_P) by a shifted error function.

ily obtainable. Note that these extensions have $\mathbf{C}^{\mathcal{N}_p}$ smoothness (in the directions normal to the immersed boundaries), because they are polynomials of order \mathcal{N}_p . The extended velocity will be called \mathbf{u}^D (since it is defined on \bar{D}), and note that \mathbf{u}^D is not necessarily periodic.

- (2) **Smoothing and periodization:** Finally \mathbf{u}^{BC} is obtained from smoothing and periodization of \mathbf{u}^D . Among some methods that can be seen in the literature (see *e.g.* [7,10,9]), we used multiplication of \mathbf{u}^D by an appropriate window function [41]

$$\mathbf{u}^{\text{BC}}(\mathbf{x}) = \varrho(\mathbf{x}) \cdot \mathbf{u}^D(\mathbf{x}), \quad (12)$$

where the window function $\varrho(\mathbf{x})$ is constructed such that it is sufficiently smooth in Ω_N and is zero in \mathcal{B} . To construct the desired window function, we used shifted and scaled error functions, exactly the same as our previous work [41].

A sectional view of an extended and periodized velocity field for linear extrapolation ($\mathcal{N}_p = 1$) is shown in Fig. 6. The physical boundary condition \mathbf{u}_{pb} was given on the physical boundary Γ_P ; and note that because of the slope of the solution in the vicinity of Γ_P (since $\mathcal{N}_P > 0$), a margin with $\mathbf{u} \neq \mathbf{u}_{\text{pb}}$ is appeared in the immersed body, which will be ignored in the solution procedure, like other immersed boundary methods.

Apparently, the computational cost of boundary condition setting is of $\mathcal{O}(\mathcal{N}_{\Omega'})$,

where $\mathcal{N}_{\Omega'}$ is the number of grid points in $\Omega' = \Omega_N \cup \mathcal{B}$. However, It should be noted that for the boundaries which are coinciding with the Cartesian grid (*e.g.* the rectangular boundaries for the Cartesian grids), the extrapolation step can be bypassed.

2.4 Filtering of the oscillations

Although the modified velocities \mathbf{u}^{BC} are $C^{\mathcal{N}_p}$ in the directions normal to the immersed boundaries, and they are periodized; however, for the finite Reynolds numbers and complex geometries there is no any guarantees for their global smoothness [41]. Moreover, the conditioned vorticity $\omega^{\text{BC}} = \nabla \times \mathbf{u}^{\text{BC}}$ is generally sharper than \mathbf{u}^{BC} . Therefore, as our numerical experiments have confirmed as well, the solution suffers from the Gibbs oscillations almost in all practical situations. Fortunately, there are some evidence which show that despite these oscillations the solutions are still accurate in the weak sense, and higher orders of pointwise accuracies can be recovered (if desired), using appropriate numerical filters [27,28,31].

In the present paper, the Helmholtz filter is used (among many kinds of the numerical filters), because of its simplicity, and since it is aimed to use it just as a *postprocessor* in presentation of the results, not for achieving the pointwise accuracies or improvement of the rates of convergence. However, our other numerical experiments (are not presented in this paper, because of lack of any theoretical support), have shown that use of this filter in the solution procedure may reduces the Gibbs oscillations.

Given the non-filtered quantity $\phi(\mathbf{x})$, the Helmholtz-filtered quantity $\bar{\phi}(\mathbf{x})$ is obtained from the following convolution

$$\bar{\phi}(\mathbf{x}) = (1 - \alpha^2 \nabla^2)^{-1} \phi(\mathbf{x}), \quad (13)$$

where α is the free parameter of the filter, standing for an appropriate length scale (see [25], or the series of works on the α modeling of turbulent flow *e.g.* [24]). Since the Laplacian operator is diagonal in the Fourier space, the above convolution takes a simple form in the wavenumber space

$$\hat{\bar{\phi}}(\mathbf{k}) = \frac{\hat{\phi}(\mathbf{k})}{(1 + \alpha^2 |\mathbf{k}|^2)}, \quad (14)$$

where \mathbf{k} is the wavenumber vector. Since the dimension of α is length, it is convenient to define a more general non-dimensional filtering factor \mathbf{C}_α such that

$$\alpha = \frac{2\pi}{L} \cdot \frac{1}{\mathbf{C}_\alpha \mathbf{k}_{\max}}, \quad (15)$$

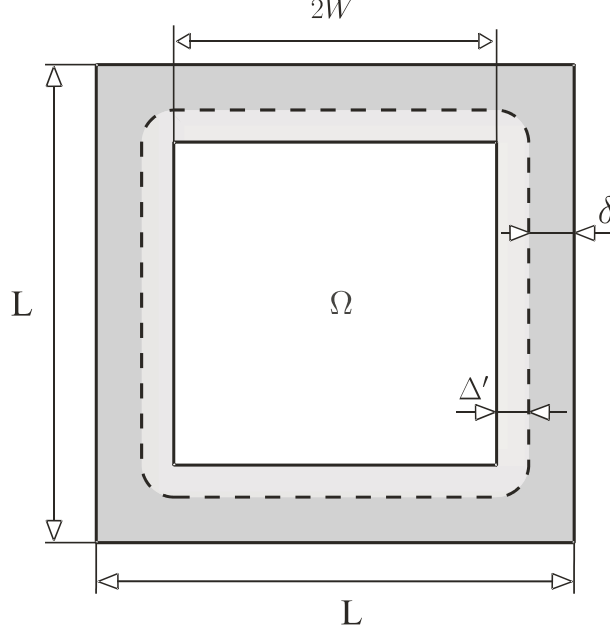


Fig. 7. Geometric parameters of the dipole–solid wall collision problem. In order to maximizing the active grid points, δ is chosen as small as possible, and different smoothness of windowing is obtained by choosing different Δ' (see also Fig. 8).

where L is the length of the solution domain, and

$$\mathbf{k}_{\max} = \|\mathbf{k}_{\max}\|_2 = \sqrt{k_{\max 1}^2 + k_{\max 2}^2} \quad , \quad (16)$$

is the maximum wavenumber, involved in the solution.

The above filter is used in the next section, wherever it is needed to remove the Gibbs oscillations, for better presentation of the results.

3 Numerical experiments

In order to assess the capabilities of the method, three test cases are analyzed in this section, including the fixed and moving boundaries, as well as given non-zero immersed velocity boundary conditions.

3.1 Dipole–solid wall collision

As a coordinate-coinciding immersed boundary problem, the recently re-analyzed dipole–solid wall collision problem [38,17] is chosen as our first numerical experiment. In a series of papers, the problem has been analyzed numerically for

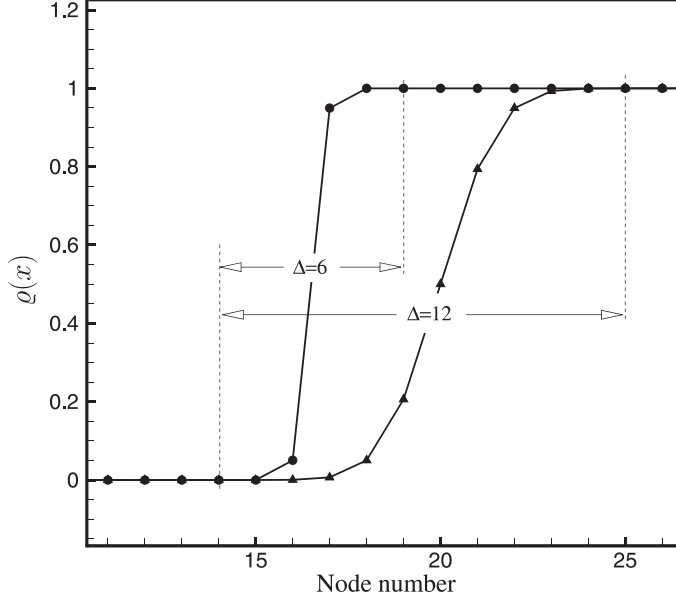


Fig. 8. A sectional view of two window functions: a fairly sharp window $\varrho_6(\mathbf{x})$ that $\Delta = \frac{\Delta'}{\Delta x} = 6$, and a moderately smooth one $\varrho_{12}(\mathbf{x})$ that $\Delta = 12$. The rising distance Δ' is the distance between the positions that $\varrho = 0$ and $\varrho = 1$ with the tolerance of $\mathcal{O}(10^{-14})$.

the normal and oblique collisions and a fairly wide range of Reynolds numbers; via finite difference, Fourier and Chebyshev spectral methods, and the volume penalization of the NSE [16,18,19,17,28]. In order to make quantitative comparisons $\mathbf{Re} = \frac{UW}{\nu} = 1000$ is chosen in this paper, similar to the Keetles *et al.* test case [28].

3.1.1 The problem setup

The geometric parameters are illustrated in Fig. 7, where it is chosen $2W = 2$ for simplicity (c.f. [28]). The margin δ should be chosen as small as possible, in order to maximizing the number of active grid points. On the other hand, it should be sufficiently wide in order to satisfy the requirements that are discussed in the appendix. Here $\delta = 10\Delta x$ is chosen (after a trial and error process) in all runs. Although fairly acceptable solutions were obtained for smaller δ , solutions exhibited more Gibbs oscillations.

As it was mentioned in §2.3.3, for the coordinate-coinciding boundaries, the extrapolation step can be bypassed, and therefore, the boundary condition setting is reduced to merely extension and windowing. In the present test case $\mathcal{N}_p = 1$ is chosen; and various numerical experiments were performed on a fairly wide range of smoothness of the window functions. The results of two typical ones, that is, a sharp window with $\Delta = \frac{\Delta'}{\Delta x} = 6$ (henceforth will be called $\varrho_6(\mathbf{x})$), and a moderately smooth one $\Delta = 12$ (which will be

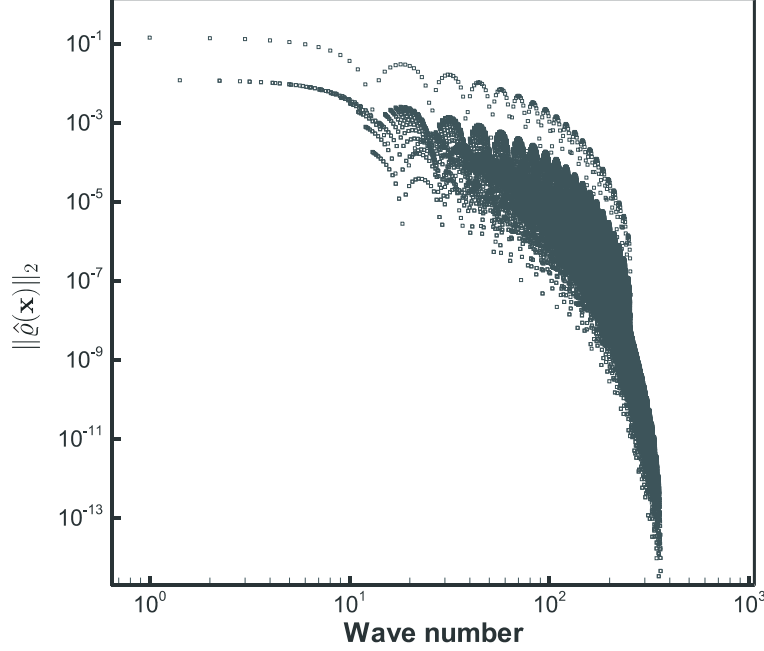


Fig. 9. Scatter diagram of the window function $\varrho_{12}(\mathbf{x})$. The exponential rate of decaying, especially in the higher wavenumber, is noticeable (c.f. [8]).

called $\varrho_{12}(\mathbf{x})$), are reported here. A sectional view of these window functions are shown in Fig. 8. It is worth mentioning that although a square, with sharp corners, is not an ideal shape for a window function; sufficiently smooth square-shaped window functions are obtainable. To emphasize this issue, the scatter diagram of the absolute values of the Fourier coefficients for $\varrho_{12}(\mathbf{x})$ is shown in Fig. 9. As one can see, the exponential rate of decaying is observable in the higher wavenumber, which can be interpreted as evidence of smoothness of the window function [8].

In the following of [28], the initial condition was constructed by summation of two identical mono-poles

$$\omega_0 = \omega_e \left[1 - \left(\frac{r}{r_0} \right)^2 \right] \exp \left[- \left(\frac{r}{r_0} \right)^2 \right], \quad (17)$$

placed at $\{(x_1, y_1), (x_2, y_2)\} = \{(0, 0.1), (0, -0.1)\}$, in which $r_0 = 0.1$, and $r = \sqrt{x^2 + y^2}$. Furthermore, to make quantitative comparisons, the total kinetic energy $\mathbb{E}(t)$, and the total enstrophy $\mathbb{Z}(t)$ are defined as

$$\mathbb{E}(t) = \frac{1}{2} \int_{\Omega} \mathbf{u}^2(\mathbf{x}, t) d\Omega; \quad \mathbb{Z}(t) = \frac{1}{2} \int_{\Omega} \omega^2(\mathbf{x}, t) d\Omega. \quad (18)$$

The main physical parameters of the initial condition are given in Tab. 1.

The solutions were performed on a 512^2 -points uniform grid and the calcu-

Table 1

The physical parameters of the initial condition of the dipole–solid wall collision problem.

ω_e	ν	U	\mathbb{E}	\mathbb{Z}	\mathbf{Re}
299.528385375226	0.001	1.0	2.0	800	1000

lations were de-aliased using the $\frac{3}{2}N$ -rule. The explicit fourth-order Runge–Kutta method with a constant timestep $\Delta t = 10^{-4}$ was used for time integration. By definition of the **CFL** number as

$$\mathbf{CFL} = \|\mathbf{u}\|_{\infty} \cdot \frac{\Delta t}{\Delta \mathbf{x}}, \quad (19)$$

in which $\|\mathbf{u}\|_{\infty}$ is the maximum velocity magnitude at each time instant, our calculations showed that $\mathbf{CFL} < 0.5$ for all times, and all runs.

3.1.2 The captured physics

The contour plots of vorticity fields at a number of time instances are presented in Fig. 10 for $\varrho_6(\mathbf{x})$ windowing. The time instances are chosen similar to Keetels *et al.* [28] for comparisons. Obviously, the no-slip and the no-penetration conditions, and all phenomena of the dipole–solid wall collision are observable (c.f. [28] or [17]). More quantitative comparisons can be made by observation of time histories of the total kinetic energy and enstrophy. In Fig. 11 these quantities are presented together with the results of Fourier–Chebyshev method and the volume penalization method (these results are extracted directly from [28]). As one can see in the left panel, decay of the total kinetic energy shows a small deviation from the Chebyshev–Fourier results after the first collision. However, this deviation is compensated in the proceeding times, such that after about $t = 1$, it is vanished. In the other words, the dissipation rates have been over-predicted during the first collision, and under-predicted for the proceeding times (in comparison to the Fourier–Chebyshev solution). In our opinion, these differences can be interpreted with regard to the smoothness of the solution (which is mainly depended on the grid resolution and sharpness of the windowing, for a fixed Reynolds number), and the order of implementation of the boundary conditions \mathcal{N}_p . In fact, the first extra-dissipation in the collision time interval is supposed to be related to appearance of the Gibbs oscillations, which transfers a part of kinetic energy into the higher wavenumber. It is this high-wavenumber energy part which will be dissipated, faster than it should, in the proceeding times. On the other hand, the under-predicted dissipation rates in the next times is presumed to be a consequence of smooth

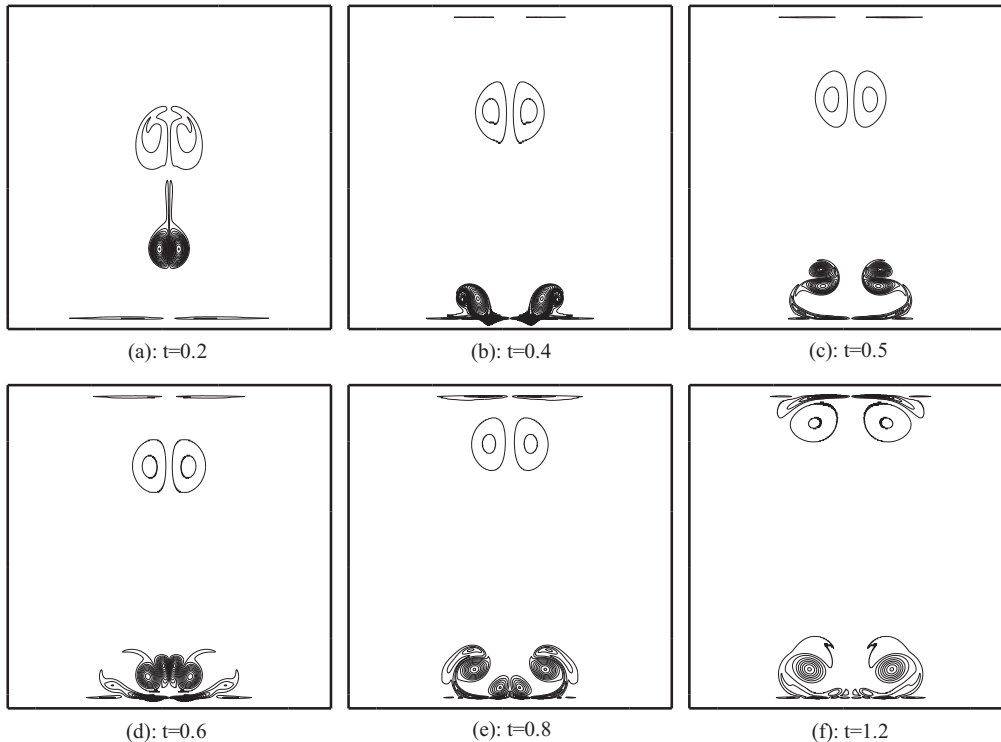


Fig. 10. Contour plots of the vorticity field for $\mathbf{Re}=1000$, obtained from application of the proposed method on a uniform 512^2 -point grid with constant timestep $\Delta t = 10^{-4}$ (amounts to $\mathbf{CFL} < 0.5$ for all times). The contour levels $(-270, \dots, -50, -30, -10, 10, 30, 50, \dots, 270)$, and the time instants are chosen similar to Keetels *et al.* [28].

walls, instead of sharp boundaries. Unfortunately, direct verification of these speculations found to be so difficult in practice, because the smoothness of the solution and the order of boundary conditions are not completely independent (see discussions in § 2.3.2 for some more details).

In addition to the results of Fourier–Chebyshev method, the results of penalized NSE are presented in Fig. 11 as well. Obviously the penalization method showed more dissipations than our method. Again we suppose that it is a consequence of sharpness of the generated boundary layers in the penalized NSE (with $\mathcal{O}(\sqrt{\epsilon\nu})$ thickness [14,28]), which is sharper than the window functions in the present method (with maximum sharpness of $\mathcal{O}(N^{-1})$). Therefore, it is not surprising that the penalization method is more dissipative than the present method in general.

On the other hand, in calculation of the total enstrophy (see the right panel of Fig. 11), this sharpness of the penalized NSE, helped the method in generation of the maximum possible vorticity in the Fourier spectral method (that is, the mean value of the vorticity in the vicinity of the discontinuities, as it is argued in [28]); while smoothness of the present method resulted in an

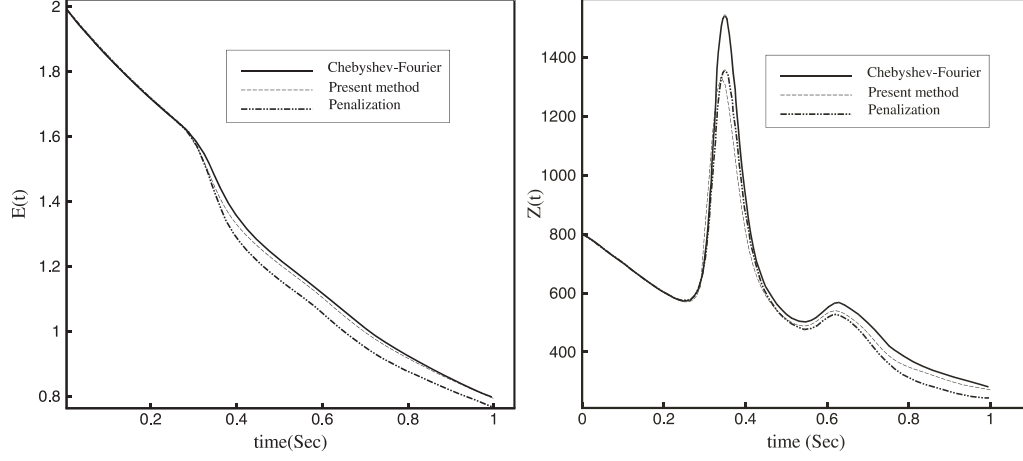


Fig. 11. Comparison of the total kinetic energy $\mathbb{E}(t)$ and enstrophy $\mathbb{Z}(t)$ of the present method with the Chebyshev-Fourier method and the volume penalized NSE [28].

under-predicted vorticity generation.

For more verification of the effects of smoothness of the windowing process, time histories of the total energy and enstrophy for $\varrho_6(\mathbf{x})$ and $\varrho_{12}(\mathbf{x})$ are compared with the penalization method in Fig. 12. Obviously the dissipation rate has changed slightly, while the vorticity generation has affected drastically, especially in the collision times.

As a conclusion, our results showed that for this grid-coinciding immersed boundaries, sharper window functions yielded better implementation of the boundary conditions, but more dissipation rates.

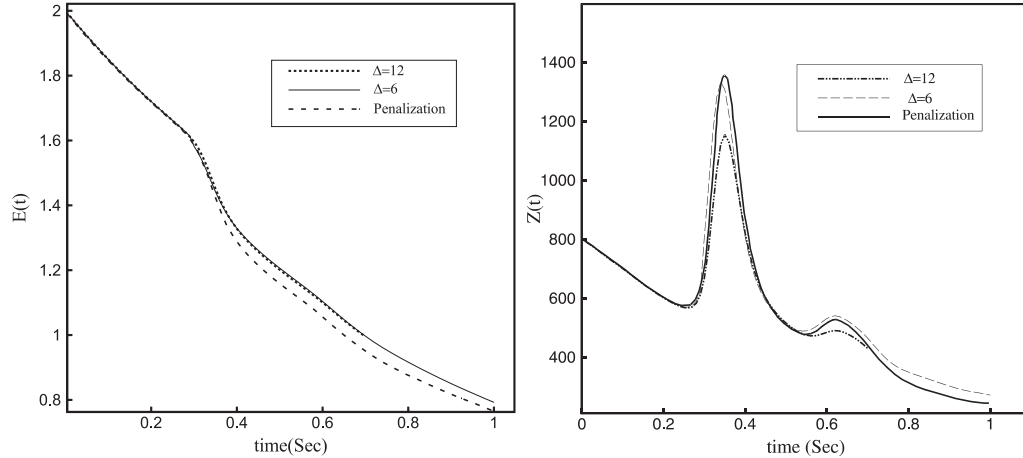


Fig. 12. Time histories of the kinetic energy and total enstrophy for two window functions (with different sharpness $\varrho_6(\mathbf{x})$ and $\varrho_{12}(\mathbf{x})$), are compared with the results of the penalization method. The sharper window function $\varrho_6(\mathbf{x})$ resulted in more dissipation and bigger total enstrophy than the smooth window function $\varrho_{12}(\mathbf{x})$.

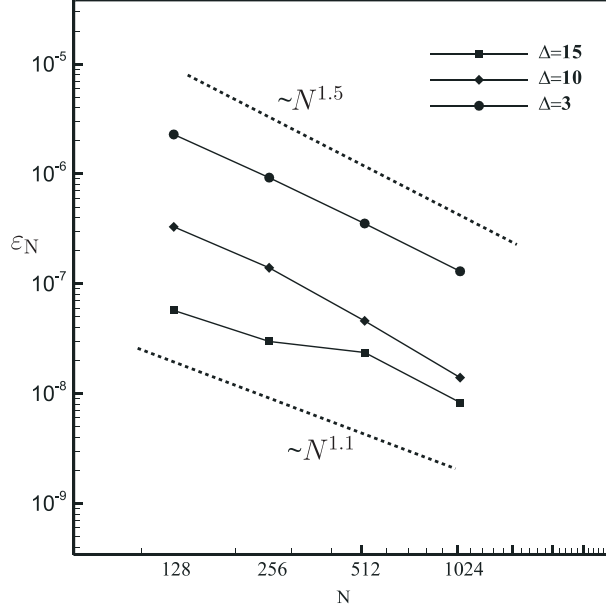


Fig. 13. The rates of grid convergence of the vorticity field for three different window functions at $t = 0.35$. The rate of convergence approaches to N^{-1} as the sharpness of the window function increase.

3.1.3 Spatial and temporal rates of convergence

It is a well-known fact that the rate of convergence of a spectral solution is highly depended on the smoothness of the solution. Particularly, for the non-smooth fields that the Fourier series are suffered from the Gibbs oscillations, the \mathcal{L}^p rate of convergence reduces to about $\mathcal{O}(1)$. Fortunately, despite this poor \mathcal{L}^p rate of convergence, it has been shown that high orders of pointwise convergence can be recovered by use of some appropriate numerical filters [27,28,31]; although the rates of convergence are depended on the order of the numerical filter this time (confirming that loss of uniform convergence does not mean necessarily loss of the accuracy in the spectral solutions). In this section, to observe the rate of convergence of the method separately, the results are reported without any kind of filtering. However, like some other spectral methods [28,31], these rates can be improved using the numerical filters.

For the initial times that the effects of the solid walls are negligible, the exponential rates for the grid convergence are easily observable for the regions far from the walls. Therefore, merely the rates of grid convergence in the collision time ($t = 0.35$ in particular) in the near-wall region are discussed here. With this regard, the rates of convergence of the vorticity field for three different window functions are presented in Fig. 13. All solutions were begun from the aforementioned initial conditions, and the solution were continued until $t = 0.35$. Similar time step sizes were used for all runs, corresponding to $\mathbf{CFL} = 0.5$ for the finest grid (that is, 2048^2 -point grid). The first-order

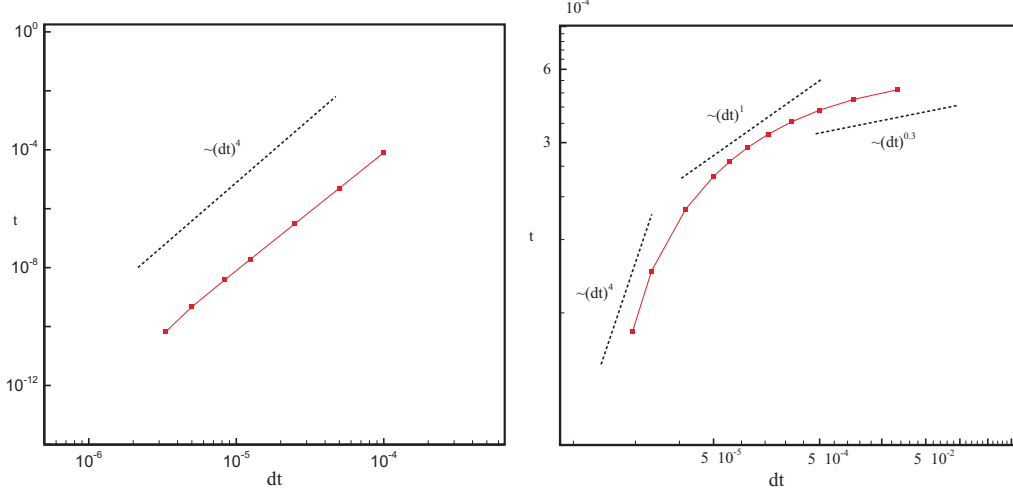


Fig. 14. The temporal rates of convergence for the no-wall flow (left), and the flow under the influence of the walls (right). The rate of convergence for wall flow is depended on the timestep sizes. However, the maximum rate of convergence of the time integration method (*i.e.*, $\mathcal{O}(dt^4)$) was achieved.

boundary condition setting $\mathcal{N}_p = 1$ was used, and the normalized \mathcal{L}^2 norms of errors

$$\varepsilon_N = \frac{\|\omega_{2048} - \omega_N\|_2}{\|\omega_{2048}\|_\infty}, \quad (20)$$

were calculated in whole of the flow domain Ω . As one can see, the rates of convergence are approached to $\mathcal{O}(N^{-1})$ by increasing the sharpness of the window functions.

In calculation of the temporal rates of convergence, and in order to reduce the Gibbs oscillations, a rather smooth window function $\varrho_{12}(\mathbf{x})$ was used on a 512^2 -point grid, and different timestep sizes. At first, we consider the temporal rate of convergence of the no-wall flow. With this regard, we examine the predicted flow at $t = 0.1$, for which the effects of the solid walls are approximately negligible. The results are presented in the left part of Fig. 14, which shows the maximum rate of convergence of the fourth-order Runge–Kutta method. Achieving the maximum rate of convergence of the time integration method, primarily shows complete decomposition of the kinematics and dynamics of the flow in the vorticity–velocity formulation of the NSE (as it was mentioned in § 2).

Then we repeated the calculations for $t = 0.4$, where the flow is under the influence of the solid wall boundary conditions. The results are presented in the right side of Fig. 14. With regard to this figure, the following points are noticeable:

- (i) Our calculations have not shown a constant rate of convergence, rather,

the rate of decaying increased by decreasing the timestep sizes. Similar behavior were observed in our other numerical experiments on different test cases and a fairly wide range of timestep sizes. It seems that this behavior of the errors is a consequence of the method of implementation of the boundary conditions, and our insistence on the explicit formulation. More or less similar non-constant temporal rates of convergence have been obtained previously in some other embedded boundary methods [21,1,2]. Although absence of a constant rate of convergence is a drawback for a method in general, achievement of the highest rates of convergence of the time integration method (that is, the fourth-order), can be seen as a merit of the method.

- (ii) Although for the sufficiently smooth solution fields the fourth order rate of convergence was obtained, our other numerical experiments have shown that for non-smooth solutions (obtained by *e.g.* sharper windowing), the Gibbs oscillations limits again the temporal rates of convergence to $\mathcal{O}(\Delta t)^m$ were $1.1 < m < 1.3$. Filtering of the solution yields higher rates which are depended mainly on the order of the filter.

3.1.4 Performance of the method

The computational costs of the present method in comparison to the classical pseudo-spectral method is given in Tab. 2. In this table, the data are presented per boundary condition setting (each Runge–Kutta sub-step). Note that the de-aliasing process (which means $\frac{3}{2}N$ zero-padding), is just needed in obtaining \mathbf{N}_1 and \mathbf{N}_2 in box (PS) in Fig. 2. Therefore, the FFTs in the boundary condition settings are performed on the original non-padded N -point grid, not on the padded grid (with the benefit of increasing the efficiency of the method). To highlight the issue, the padded and non-padded FFTs are separated in Tab. 2. As one can see, implementation of the immersed boundaries needs just four non-padded FFTs, for each sub-step. The CPU times and required memories are obtained for a 3GHz Intel Pentium 4 system. Both methods used the same FFT subroutine and optimization levels, and the CPU times are in

Table 2

Computational costs for the classical pseudo-spectral (CPS), and the proposed Fourier immersed boundary method (FIBM); for each sub-step of fourth-order Runge–Kutta method. The FFTs are divided into the FFTs on the padded domain (P), and the non-padded domain (N–P).

Method	N	#FFT (P)	#FFT (N–P)	CPU time	Memory
FIBM	512^2	16	4	0.947 Sec.	105 MB
CPS	512^2	16	0	0.905 Sec.	105 MB

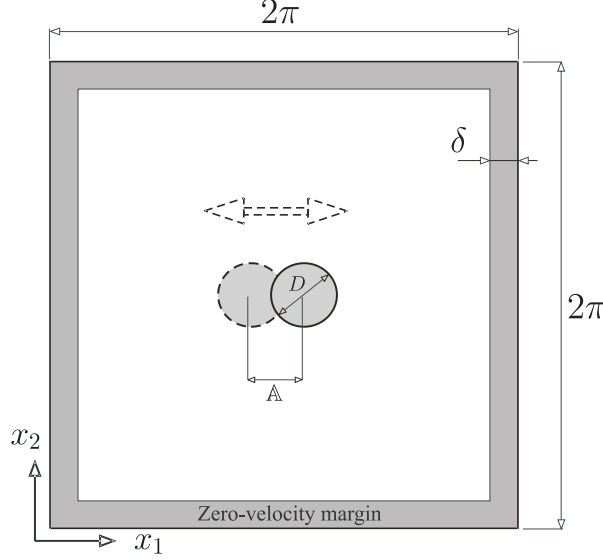


Fig. 15. Geometric parameters of the oscillating circular cylinder problem

average. Apparently, the method is really efficient with less than 5% additional CPU time and without any sensible increasing in the required memory, in comparison to the Fourier pseudo-spectral method. Such a performance, in addition to robustness of the windowing procedure, makes the method ideal for the moving boundary, multi-object... problems, which some examples can be seen in the sequel.

3.2 Oscillating circular cylinder in fluid at rest

Undoubtedly, one of the most attractive features of the immersed boundary methods is the possibility of simple and efficient treatment of the complex and moving boundary problems. In the present section, to show the capability of the method in simulation of the moving boundary flow, the problem of in-line oscillating cylinder in a quiescent fluid is examined.

3.2.1 The problem setup

The problem has been analyzed several times experimentally [22], as well as numerically [30,46,33], in a fairly wide range of involved physical parameters; which are the Reynolds number $\mathbf{Re} = UD\nu^{-1}$, and the Keulegan–Carpenter number $\mathbf{KC} = U(fD)^{-1}$. In these definitions, U is the maximum velocity, f is the oscillations frequency, and D is the cylinder diameter. In the present study, to facilitate assessment of the results, the well-documented combination $(\mathbf{Re}, \mathbf{KC}) = (100, 5)$ is considered, which the previous experiments have shown that two-dimensional calculations are rather justifiable [22].

Table 3

The physical as well as numerical parameters, used in the solution of the oscillating cylinder problem.

D	f	\mathbb{A}	ν	$\delta/\Delta x$	\mathbf{N}	Δt	\mathbf{CFL}
0.35	1.0	0.27852	6.1249747×10^{-3}	8	512^2	10^{-4}	0.2

A simple harmonic oscillation in the x_1 direction is exerted to the cylinder center as

$$x_{1c}(t) = \mathbb{A} \sin(2\pi ft), \quad (21)$$

where \mathbb{A} is the amplitude of the oscillations. It can easily be shown that for such harmonic oscillations $\mathbf{KC} = 2\pi\mathbb{A}D^{-1}$. Fig. 15 illustrates the problem setup, and Table (3) summarizes the main physical and numerical parameters of the solution.

The problem was solved on a fixed 512^2 -point equi-spaced Cartesian grid, distributed on a $(2\pi)^2$ regular domain, using a fixed timestep size $\Delta t = 2 \times 10^{-4}$ Sec., equivalent to $\mathbf{CFL} \approx 0.2$. The zero velocity margin $\delta = 8\Delta x$ was chosen, which resulted in 504^2 active grid points. On such a grid, there were about 28 grid points across the cylinder diameter, and the number of numerical boundary points was about 30 in average.

To keep the solution as smooth as possible, the solution was started from zero velocity at $\theta = -\frac{\pi}{2}$ phase angle, and therefore, a 0.25 Sec. time shifting is used when comparing our results with the other data. Although the no-slip conditions are not satisfied accurately in the initial time steps (as discussed in § 2.2, and also see [47,45]), it developed appropriately in the next times, and therefore the results for $\mathcal{N}_r = 1$ are presented here.

3.2.2 The captured dynamics

The problem was solved for three different orders of boundary conditions (*i.e.*, $\mathcal{N}_p = 0, 1, 2$), and the effects of \mathcal{N}_p will be discussed later. Before that, the captured dynamics for $\mathcal{N}_p = 2$ is presented here.

The obtained vorticity dynamics is presented in Fig. 16. As one can see, it is in good agreement with the reported experimental and numerical studies of this $(\mathbf{Re}, \mathbf{KC})$ combination (c.f. [22,46,33]). As the cylinder moves to one side, upper and lower boundary layers develop with symmetric starting and separation points; while a wake develops in the downstream of the cylinder, containing two symmetric counter-rotating vortices. Then in the opposite side cycle similar structures are observable, and the last wake vortices (now located in front of the cylinder), split by the cylinder. More quantitative comparisons

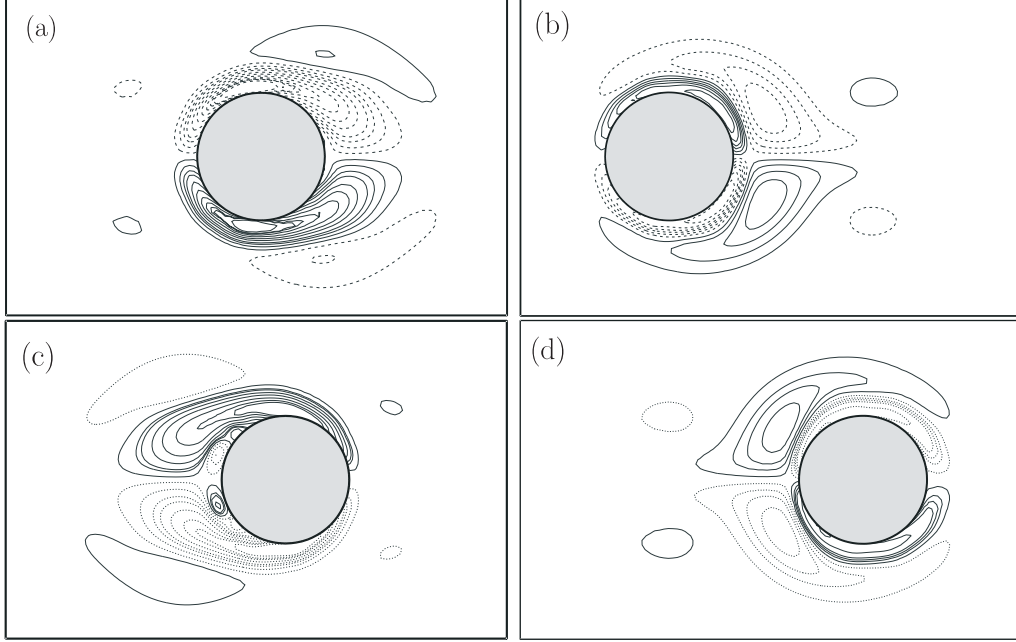


Fig. 16. Vorticity iso-lines of four phase angles (a): 0° , (b): 96° , (c): 192° , (d): 288° . The Helmholtz filter (with $\mathbf{C}_\alpha = 0.46$) is used as a post-processor in order to remove the Gibbs oscillations (see § 2.4). The dashed lines denote negative values.

can be made by comparing the instantaneous velocity profiles, as illustrated in Fig. 17. The velocities are compared with the experimental data of Dautch *et al.* [22], and similar to some other works [22,46,33], four x_1/D sections are illustrated for each of the three phase angles. As one can see, there is a general agreement between the present results and the experimental data.

It should be noted that because of high levels of oscillations (especially for the vorticity fields), the aforementioned Helmholtz filter (see § 2.4) with $\mathbf{C}_\alpha = 0.46$ was used (as a post-processor), in obtaining the vorticities of Fig. 16. The \mathbf{C}_α was chosen by observation, such that the active contour levels can be compared with the benchmark data. Obtaining fairly acceptable results from an oscillating solution by use of a simple filter (just as a post-processor) is interesting, and it may be interpreted as a confirmation of the previous observations (*e.g.*, [27,28,31]) that the higher degrees of pointwise accuracies can be recovered from a contaminated spectral solution (at least in some circumstances).

3.2.3 On the effects of order of the boundary condition setting

Presence of various numerical and experimental studies of this problem offers a unique opportunity for direct evaluation of the effects of order of implementation of the boundary conditions (*i.e.*, \mathcal{N}_p), on the overall accuracy of

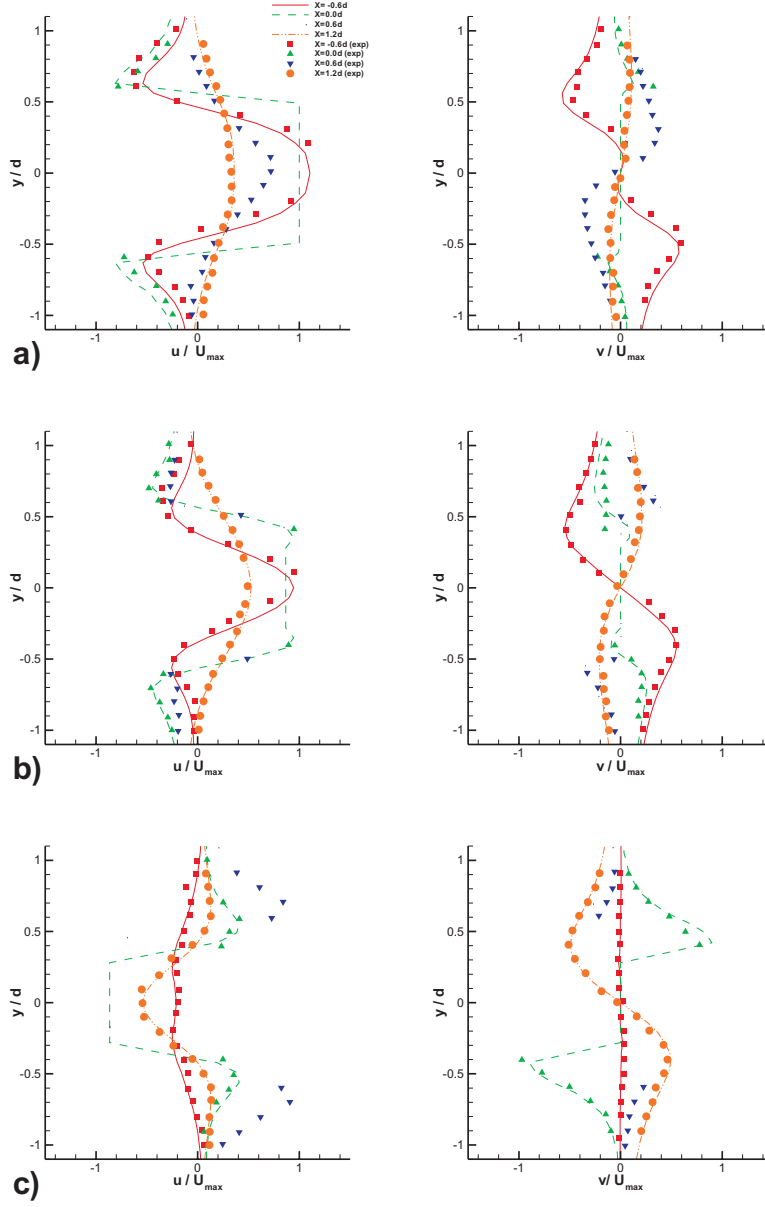


Fig. 17. Comparison of the instantaneous velocity profiles for three phase angles (a): 180° , (b): 210° , and (c): 330° . Lines show the results of the present method (obtained from the second-order boundary conditions $\mathcal{N}_p = 2$), while the symbols are the experimental data [22].

the solution. In the last test case (*i.e.*, the dipole–wall collision), the \mathcal{L}^p convergence of the vorticity fields were considered, which were highly under the influences of the Gibbs oscillations (see §3.1.3, and Figs. 13 and 14). Instead, in the present section, it is aimed to evaluate the accuracy of the solution by direct comparison of the velocities with a second-order solution, which has been verified several times (that is, the Dutsch *et al.* solution [22]).

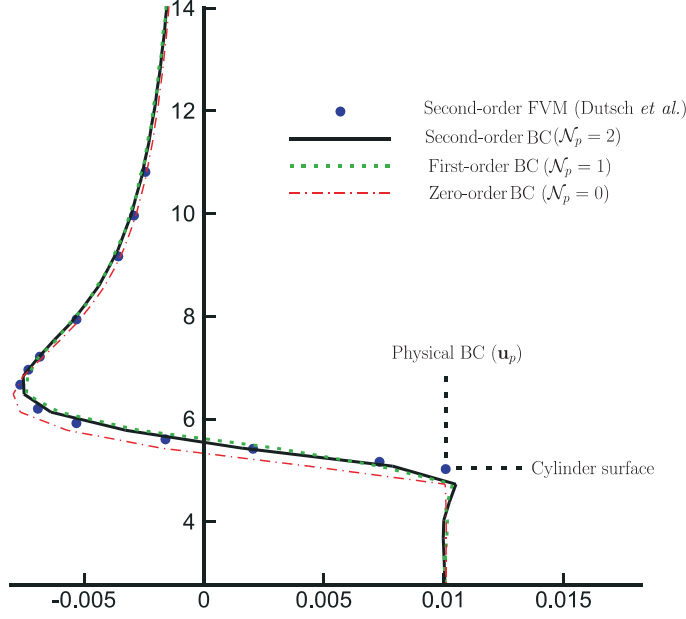


Fig. 18. Longitudinal velocity profiles at $x_1/D = 0$ of the 0° phase angle. The results of different \mathcal{N}_p are compared with the second-order finite volume solution of Dutsch *et al.* [22].

In Fig. 18 the longitudinal velocity for different orders of implementation of the boundary conditions (that is, $\mathcal{N}_p=0,1,2$), are compared with the second-order finite volume solution of Dutsch *et al.* [22] in the $x/D = 0$ section of the 0° phase angle. This position is chosen intentionally because the effects of accuracy of the longitudinal velocity boundary condition can be evaluated directly. The following points can be made with regard to this figure:

- i) As one can see, the first-order and second-order implementation of the boundary conditions $\mathcal{N}_p = 1, 2$, yielded velocity profiles that are very close to the second-order finite volume solution. Therefore, we can say that for these orders of boundary conditions, the solution has been practically equivalent to a second-order solution. Comparisons of the velocities in the other phase angles and other x/D sections (are not presented here for the sake of brevity), do confirm the above statement.
- ii) Even for the least order of boundary conditions (that is $\mathcal{N}_p = 0$), except for the regions near to the cylinder surface, the velocities are again really close to the second-order finite volume solutions. This issue is practically important, since it justifies use of easy-to-implement case $\mathcal{N}_p = 0$ (which bypasses some tedious geometric operations), in rough solutions for even very complex geometries.
- iii) Our numerical experiments have shown that the minimum amounts of the Gibbs oscillations occurs for $\mathcal{N}_p = 0$, and the maximum occurs for $\mathcal{N}_p = 2$. Note that the velocities are presented here without any kind of filtering. In fact, just for presentation of the vorticity fields (in Fig. 16)

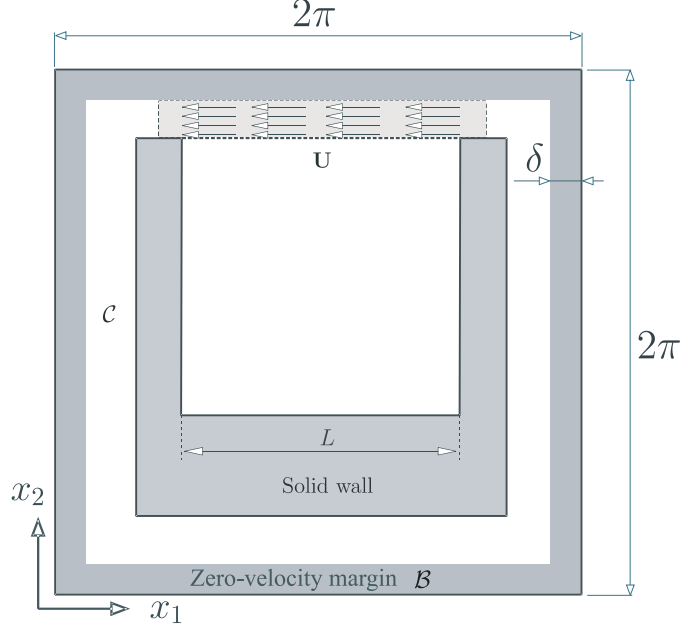


Fig. 19. Geometry of the lid-driven cavity problem. The side channel \mathcal{C} is considered in order to balance the mass flow rate and achieving zero mean velocity and vorticity.

the Helmholtz filter is used, because of high levels of the Gibbs oscillations.

3.3 Lid-driven cavity flow

The method is not restricted to the no-slip and no-penetration immersed boundary conditions. In fact, any arbitrary immersed velocity boundary condition (which can be presented by a window or mask function), can be implemented via the proposed algorithm. To show this capability of the method, the classical two-dimensional lid-driven cavity flow is investigated in two different regimes. For the steady solutions, a fairly low Reynolds number flow $\mathbf{Re}=100$ is chosen, while the unsteady solutions is examined by analysis of a higher Reynolds number flow, that is $\mathbf{Re}=1000$. Table 4 presents the physical and numerical characteristics of the solutions.

Our suggested configuration is illustrated in Fig. 19. As one can see, the solid walls of cavity are implemented by a U-shaped solid immersed body, and a side channel (denoted by \mathcal{C}), is considered in order to balance the overall mass flow rate, and achieving zero mean velocity and vorticity fields. Both no-slip conditions and given velocity \mathbf{U} are set at once in the beginning of each time step, as explained in § 2.2. Since the cavity walls were coinciding with the Cartesian grid, the numerical and physical boundary points are coinciding, and therefore, extrapolation process is bypassed (similar to the dipole–solid

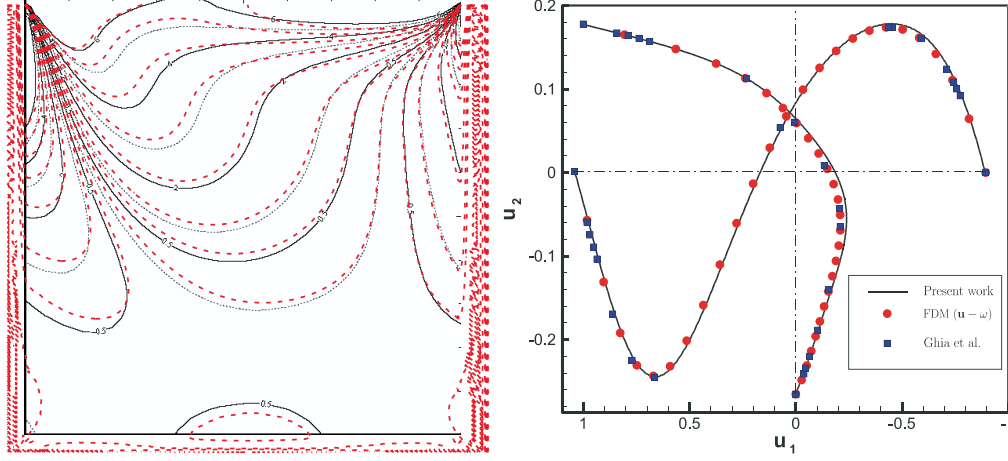


Fig. 20. **Left:** The vorticity iso-lines of the steady solution ($\mathbf{Re}=100$) in comparison to the results of Auteri *et al.* [6]. The dashed lines show the results of the present work. **Right:** The velocity profiles in comparison to the results of Ghia *et al.* [26] and a classical second-order finite difference vorticity-velocity solution.

wall problem).

At first we examine the steady solution. Several numerical and experimental studies have demonstrated presence of steady solution for $\mathbf{Re} = \mathbf{UL}/\nu = 100$ (see *e.g.* [26,6,5,36]). In the present work, the problem was solved on a 256^2 -point grid, which because of presence of the margin \mathcal{B} , the side channel \mathcal{C} , and the solid U-shaped body, there were 185^2 active grid points inside the cavity. Since the computational cost of the method scales by $(N \log N)$, one can said that about 48% extra cost was paid for a 185^2 -point simulation. The solution begun from zero velocity field, and a constant time step $\Delta t = 2 \times 10^{-3}$ Sec. (equivalent to $\mathbf{CFL} \approx 0.5$) was used; and the solution was continued until the steady solution was established after about 8.6 Sec.

On the other hand, the numerical experiments showed that $\mathcal{N}_r > 1$ was needed for adequate satisfaction of the no-slip conditions. It can be a consequence of dominance of the solid walls on the solution in this particular flow. In fact, when implementing the boundary conditions, a huge amount of discontinuities are imposed to the \mathbf{u}^{BC} (in comparison to *e.g.*, the oscillating cylinder problem). Therefore, removing these discontinuities, and enforcing the solenoidality, changes the immersed boundary conditions of $\mathbf{u}_{\text{sol}}^{\text{BC}}$ such that it

Table 4

Physical and numerical parameters of the cavity flow.

	\mathbf{Re}	Grid resolution	Active grid resolution	Δt	\mathbf{CFL}
steady	100	256^2	185^2	2×10^{-3}	≈ 0.5
unsteady	1000	512^2	375^2	10^{-4}	≈ 0.5

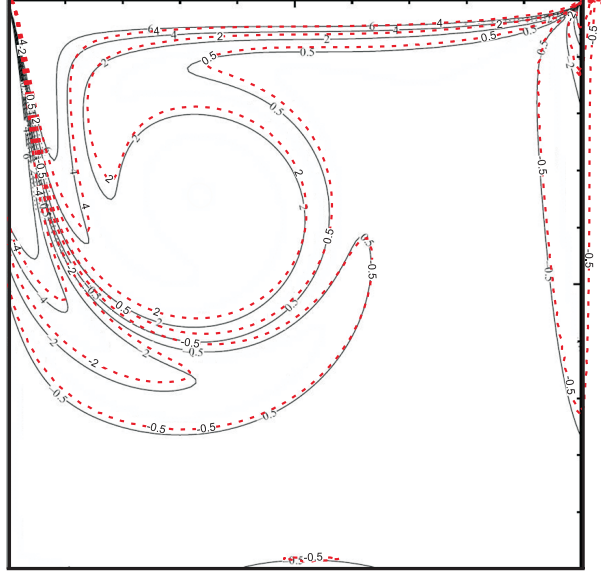


Fig. 21. The vorticity iso-lines of the unsteady solution for $\mathbf{Re}=1000$, at the time instant $t = 6.25$ Sec, compared with the data of Auteri *et al.* [6]. The dashed lines show the results of the present method.

is needed to set them more than once (see § 2.2). However, our experiments did not show changes in the results for $\mathcal{N}_r > 3$, and therefore, the present results are obtained with $\mathcal{N}_r = 3$.

The results are shown in Fig. 20. In the left panel, the vorticity iso-lines are compared with the results of singularity-removed Chebyshev solution of Auteri *et al.* [6]. As one can see, there is a good overall agreement between the results, although some discrepancies (especially in the middle of the cavity) are noticeable. However it should be emphasized that there are substantial differences between our solution and the Auteri *et al.* method [6], in implementation of the boundary conditions and levels of solenoidality. In the right side of Fig. 20, the velocity profiles in the middle of cavity are presented and compared with the data of Ghia *et al.* [26], and a classical second-order finite difference solution of vorticity-velocity formulation of the NSE. In the finite difference solution, the no-slip conditions are implemented via the explicit Thom's rule [40], which satisfies just approximately the solenoidality [39]. As it can be seen, while the Ghia *et al.* data (which obtained from solution of the primitive variables formulation of the NSE, and pure Neumann boundary conditions for the Poisson pressure equation), are coinciding with the finite difference results; they show some discrepancies with our data, especially far from the solid walls.

In addition to the steady solution, it is aimed to verify time accuracy of the method. To this end, a higher Reynolds number (that is, $\mathbf{Re}=1000$) in an unsteady regime is examined. The unsteady cavity flow has studied several times experimentally and numerically (see *e.g.* [6,5,36]). The problem was solved on a 512^2 -point equi-spaced Cartesian grid which resulted in 375^2 active

grid points inside of the cavity. Therefore, the extra cost of the calculations has been about 46%. To keep the **CFL** number about 0.5, a constant time step $\Delta t = 10^{-4}$ was chosen. The solution was begun from a zero velocity field at $t = 0$, and continued until $t = 6.25$, which the results are comparable by the other results [6,5]. Like the last test case, $\mathcal{N}_r = 3$ was used. In Fig. 21 the vorticity iso-lines in comparison to the results of singularity-removed Chebyshev solution [6] are shown. Good agreements between the results are observable, despite the fact that a different time integration method (that is, the second-order Adams–Bashforth method) is used in obtaining the results of [6] and [5].

4 Conclusions

An immersed boundary Fourier pseudo-spectral method was proposed for the vorticity-velocity formulation of the two-dimensional incompressible NSE. The zero-mean Fourier pseudo-spectral solution are used, and therefore, the method is applicable to the confined flows, which are modeled by considering a zero velocity margin in the vicinity of the regular boundaries. Without explicit addition of external forcing functions, arbitrary Dirichlet velocity boundary conditions are implemented by direct modification of the diffusion and convection terms of the vorticity transport equation; and in this way, it was shown that the obtained velocities are solenoidal. The immersed boundary conditions are approximated on some regular grid points (called the numerical boundary points), by use of different orders (up to second-order) polynomial extrapolations, in the normal directions to the immersed surfaces.

Although because of presence of the Gibbs phenomenon the \mathcal{L}^p spatial rates of convergence are not so high (even less than two, especially in the regions near the solid walls), good agreements between our results with some other second-order solutions were observed in some test cases. On the other hand, the temporal rate of convergence was found to be a function of timestep sizes, and for sufficiently small timestep sizes, the highest order of time discretization (that is, the fourth order) was obtained.

Like many other immersed boundary methods, it was observed that implementation of the immersed velocity boundary conditions imposes some discontinuities; and conversely, imposing the continuity changes the immersed boundary conditions. However, it was observed that by repeating the procedure, and by development of the solution, some solenoidal velocities, which satisfy the immersed boundary conditions (approximately) are obtainable.

The method was implemented to a conventional Fourier pseudo-spectral solver of the vorticity-velocity form of the NSE, without any changes in the internal structure of the solver, and the performance of the method was compared with the original solver. In this context, it was observed that fixed and moving

immersed boundaries can be treated by a computational cost, which scales by $(N \log N)$.

The best accuracies (and also the most Gibbs oscillations), were obtained for the second-order boundary conditions. However, even for the easy-to-implement zero-order boundary conditions, fairly accurate solutions were obtained, at least far from the solid walls. Particularly this feature of the method is interesting, because this simple algorithm can be used in a rather wide range of applications, from rough simulations to the fairly accurate ones, just by changing the order of implementation of the boundary conditions. In our opinion, it is particularly a consequence of using a pseudo-spectral solver as the core of the method.

Just a simple constant-width Helmholtz filter was used in the present work, as a postprocessor, and in order to remove the Gibbs oscillations in visualization of the results. However, like other spectral methods, one can use some particular numerical filters to improve the rates of convergence. This is one of several lines which we can assume for the extension of the present method.

Appendix: Some fundamental considerations about the proposed algorithm

Since the zero-mean Fourier series are used in the present method, the method is applicable to the flows with zero-mean velocity and vorticity, and with zero dynamics of the mean vorticity (see Eqns. (4) or (8)). Now from the Stokes theorem, the mean vorticity reads

$$\hat{\omega}(0,0) = \bar{\omega} = \frac{1}{\Omega_D} \oint_{\Gamma_D} \mathbf{u}(\Gamma_D) \cdot d\Gamma_D. \quad (.1)$$

Among many flow configurations with $\bar{\omega} = 0$, the ones that $\mathbf{u}_{\Gamma_D} = 0$ are dealt in the present work, and these flows are modeled by considering a zero-velocity margin, as it is shown in Fig. 1.

With this in mind, the present appendix is devoted to explanation of three fundamental issues, particularly related to steps 3 and 4 of the proposed algorithm (§ 2.2). These discussions are presented here separately for the sake of clarity, and to prevent disturbing the main flow of the paper.

.1 The mean value of the conditioned vorticity

Clearly, Eq. (6) yields a zero-mean conditioned vorticity (since $k_1 = k_2 = 0$ results in $\hat{\omega}^{BC}(0,0) = 0$), as it is desired. However, legitimacy of application of this equation for arbitrary \mathbf{u}^{BC} (obtained from arbitrary immersed boundary conditions and different window functions), needs some considerations.

In general, the Stokes theorem states that for a confined flow ($\mathbf{u}_{\Gamma_D} = 0$) the mean vorticity is vanished, regardless of distribution of velocities in D (see Eq. .1). However, the point is that the Stokes theorem is applicable to the velocity fields that are C^1 at least; the condition which can be violated in the general immersed boundary condition settings and windowings.

On the other hand, note that it is desired to find the solution in the Fourier space, and regardless of rate of decaying of $\hat{\mathbf{u}}^{BC} = \mathbf{FT}\{\mathbf{u}^{BC}\}$ (which is depended on the smoothness of \mathbf{u}^{BC}), however, $\mathbf{FT}^{-1}\{\hat{\mathbf{u}}^{BC}\}$ has C^∞ smoothness (since it is constructed from the Fourier functions which are C^∞ smooth). Now, by applying the Stokes theorem on $\mathbf{FT}^{-1}\{\mathbf{FT}\{\mathbf{u}^{BC}\}\}$, one can find

$$\bar{\omega}^{BC} = \frac{1}{\Omega_D} \int_{\Omega_D} (\nabla \times \mathbf{u}^{BC}) d\Omega_D = \frac{1}{\Omega_D} \oint_{\Gamma_D} \mathbf{FT}^{-1}\{\mathbf{FT}\{\mathbf{u}^{BC}\}\}(\Gamma_D) d\Gamma_D. \quad (.2)$$

On the other hand, it should be noted that $\mathbf{FT}\{\mathbf{u}^{BC}\}$ can be suffered from the Gibbs oscillations because of presence of discontinuities in \mathbf{u}^{BC} , and therefore, $\mathbf{FT}^{-1}\{\mathbf{FT}\{\mathbf{u}^{BC}\}\}$ is not necessarily zero on Γ_D . Consequently, $\bar{\omega}^{BC}$ is not vanishing in general.

However, since the zero-mean flows are dealing, these $\bar{\omega}^{\text{BC}}$ supposed to be discrepancies from the flow physics, and they are ignored in the solution procedure. On the other hand, as our numerical experiments were shown, these spurious mean vorticities decrease as the solution is developed; and for the logical timestep sizes, they get practically-ignorable values almost immediately after beginning of the solution.

.2 On the solenoidality of the velocities

A key feature of the present method is that the physical (solenoidal) velocities are obtained and used, regardless of complexity of the immersed boundary conditions. Here it is desired to justify why Eq. (7) results in solenoidal velocities for an arbitrary ω^{BC} .

Proposition 1 *For any conditioned vorticity $\omega^{\text{BC}} \in \mathcal{L}^2(\bar{D})$ the velocity vector*

$$\hat{\mathbf{u}}_{\text{Sol}}^{\text{BC}} = -i \frac{\mathbf{k}^\perp}{|\mathbf{k}|^2} \hat{\omega}^{\text{BC}} \quad (.3)$$

is solenoidal, in which $\hat{\omega}^{\text{BC}} = \mathbf{FT}\{\omega^{\text{BC}}\}$.

Note that Eq. (7) is re-written again (that is Eq. .3) for clarity.

PROOF. There are many ways to proof the above proposition, and we give one of them which is more consistent with our purposes.

At first, since $\omega^{\text{BC}} \in \mathcal{L}^2(\bar{D})$, the Fourier coefficients $\hat{\omega}^{\text{BC}}(\mathbf{k}) = \mathbf{FT}\{\omega^{\text{BC}}\}$ exist for any \mathbf{k} , although their rate of decaying can be not so high. Now, it is a well-known fact [13] that the continuity equation in the Fourier space reduces to

$$\mathbf{k} \cdot \hat{\mathbf{u}}_{\text{Sol}} = 0. \quad (.4)$$

In the other words, the solenoidal velocities are perpendicular to the wavenumber vector and vice versa. Now referring to Eq. (.3), the proposition is proven since the velocities $\hat{\mathbf{u}}_{\text{Sol}}^{\text{BC}}$ are obtained from \mathbf{k}^\perp , and therefore they are perpendicular to \mathbf{k} . ■

In addition to the above proof, also our extensive numerical experiments on different test cases have shown that the divergence of $\mathbf{u}_{\text{Sol}}^{\text{BC}}$ remains in the order of machine accuracy everywhere in \bar{D} .

The above simple proof in the Fourier space should not hide the detailed mechanism that discards the divergence. In fact, by taking the divergence of both sides of Eq. (2) we have

$$\nabla^2(\nabla \cdot \mathbf{u}) = \nabla \cdot (\hat{\mathbf{e}}_z \times \nabla \omega) = 0, \quad (.5)$$

which means the obtained velocities from Poisson equation Eq. (2) are such that the divergence of them is a harmonic function (and therefore gets its maximum values on the boundary Γ_D), regardless of the boundary conditions. As an immediate consequence, the divergence of the velocities can be controlled by controlling the divergence on (near) the boundary Γ_D . This statement is not new, and it has been mentioned and discussed previously in many references (see *e.g.* [20] or [39]).

Now definition of a zero velocity margin \mathcal{B} in the vicinity of Γ_D can be seen as a way for controlling the divergence. This strategy is not restricted to the spectral method, and can be used in other numerical methods, *e.g.* the finite difference method. However, in the Fourier pseudo-spectral solutions, since there is not any boundaries, the divergence vanishes perfectly everywhere on D , as it proved. In these cases the margin \mathcal{B} helps achieving the higher rates of decaying of the Fourier coefficients.

In fact, we found that considering a closed flat zero velocity margin in the vicinity of the regular boundary (when it is possible), is a good idea for overcoming many difficulties related to finding appropriate vorticity boundary conditions and obtaining the solenoidal velocities. We will follow this idea in our next works on the fluid–solid interaction problems.

.3 About the dynamics of the mean vorticity

There is still another question that should be answered with regard to the proposed algorithm. In fact, by time integration of Eq. (8), we assumed zero dynamics for the mean vorticity during the time interval Δt . Legitimacy of this assumption is in order.

Note that in modeling of the flow configuration of Fig. 1, the modified \mathbf{L} , \mathbf{N}_1 , and \mathbf{N}_2 are introduced into the vorticity transport equation, and then this modified equation is transformed into the Fourier space (yielded Eq. (8)). Therefore, in order to analysis of the dynamics of the mean vorticity of our numerical model, the mean mode of the modified vorticity transport equation should be analyzed.

Theorem 1 *Given a conditioned vorticity $\omega^{\text{BC}} = \nabla \times \mathbf{u}^{\text{BC}}$, time integration*

of

$$\partial_t \omega = \nu \nabla^2 \omega^{\text{BC}} - \frac{\partial^2}{\partial x_1 \partial x_2} (u_2^2 - u_1^2)_{\text{Sol}}^{\text{BC}} + \left(\frac{\partial^2}{\partial x_2^2} - \frac{\partial^2}{\partial x_1^2} \right) (u_1 u_2)_{\text{Sol}}^{\text{BC}}, \quad (.6)$$

results in zero dynamics for the mean vorticity, in which $\mathbf{u}_{\text{Sol}}^{\text{BC}} = \mathbf{F}\mathbf{T}^{-1}\{\hat{\mathbf{u}}_{\text{Sol}}^{\text{BC}}\}$, and $\hat{\mathbf{u}}_{\text{Sol}}^{\text{BC}}$ is obtained from Eq. (7).

PROOF. To show $\partial_t \bar{\omega} = 0$, one way is to show that the modified \mathbf{L} , \mathbf{N}_1 , and \mathbf{N}_2 have zero means. Now, at first, note that the diffusion term \mathbf{L} (that is, the Laplacian of vorticity) is essentially silent about the mean values, since it consists of the (second-order) derivatives in both directions. On the other hand, the mean values of the other two convection terms (that is, \mathbf{N}_1 , and \mathbf{N}_2) can be found directly. For example

$$(\bar{\mathbf{N}}_1)_{\text{Sol}}^{\text{BC}} = \frac{1}{l_1 l_2} \mathcal{D}_{x_1}^{-1} \mathcal{D}_{x_2}^{-1} (\mathbf{N}_1)_{\text{Sol}}^{\text{BC}} = \frac{1}{l_1 l_2} \mathcal{D}_{x_1}^{-1} \mathcal{D}_{x_2}^{-1} [\mathcal{D}_{x_1} \mathcal{D}_{x_2} (u_2^2 - u_1^2)_{\text{Sol}}^{\text{BC}}], \quad (.7)$$

where \mathcal{D}_{x_i} stands for derivative with respect to x_i ; while $\mathcal{D}_{x_i}^{-1} = \int_0^{l_i} dx_i$ shows the inverse of derivative operator \mathcal{D}_{x_i} (that is, integration on the regular domain \bar{D}). Now, implementation of the integral operators and changing the order yields

$$\begin{aligned} (\bar{\mathbf{N}}_1)_{\text{Sol}}^{\text{BC}} &= \frac{1}{l_1 l_2} (\mathcal{D}_{x_1}^{-1} \mathcal{D}_{x_1}) [\mathcal{D}_{x_2}^{-1} \mathcal{D}_{x_2} (u_2^2 - u_1^2)_{\text{Sol}}^{\text{BC}}] \\ &= \frac{1}{l_1 l_2} (\mathcal{D}_{x_1}^{-1} \mathcal{D}_{x_1}) [(u_2^2 - u_1^2)_{\text{Sol}}^{\text{BC}} |_0^{l_2}] \\ &= \frac{1}{l_1 l_2} [(u_2^2 - u_1^2)_{\text{Sol}}^{\text{BC}} |_0^{l_2}] |_0^{l_1}. \end{aligned} \quad (.8)$$

On the other hand, since $\mathbf{u}_{\text{Sol}}^{\text{BC}}$ is double periodic on \bar{D} , one can write

$$\begin{aligned} (u_2^2 - u_1^2)_{\text{Sol}}^{\text{BC}}(x_1 = 0, x_2) &= (u_2^2 - u_1^2)_{\text{Sol}}^{\text{BC}}(x_1 = l_1, x_2), \\ (u_2^2 - u_1^2)_{\text{Sol}}^{\text{BC}}(x_1, x_2 = 0) &= (u_2^2 - u_1^2)_{\text{Sol}}^{\text{BC}}(x_1, x_2 = l_2). \end{aligned} \quad (.9)$$

Substitution of these relations into Eq. (.8), yields $(\bar{\mathbf{N}}_1)_{\text{Sol}}^{\text{BC}} = 0$. Exactly the same procedure can be followed for the $(\bar{\mathbf{N}}_2)_{\text{Sol}}^{\text{BC}}$ which results in

$$(\bar{\mathbf{N}}_2)_{\text{Sol}}^{\text{BC}} = 0. \quad (.10)$$

■

Remark 2 Note that changing the derivatives order from Eq. (.7) to Eq. (.8) needs $\mathbf{u}_{\text{Sol}}^{\text{BC}} \in C^1(\bar{D})$ at least. On the other hand, as it was argued before, $\mathbf{u}_{\text{Sol}}^{\text{BC}} = \mathbf{F}\mathbf{T}^{-1}\{\mathbf{F}\mathbf{T}\{-\nabla \times \omega^{\text{BC}}\}\} \in C^\infty(\bar{D})$ regardless of the rate of convergence of $\hat{\mathbf{u}}_{\text{Sol}}^{\text{BC}}$.

Remark 3 Although the theorem and proof are offered for this particular form of the vorticity transport equation, that is Eq. (3); it is worth mentioning that they are valid and applicable to the conventional form of vorticity transport equation (1), but it is just needed applying the Gauss theorem once, on each convection term.

References

- [1] S. Abarbanel, A. Ditkowski, B. Gustafsson, On error bounds of finite difference approximations to partial differential equations— temporal behavior and rate of convergence, J. Sci. Comput. , Vol. 15, No. 1, 2000.
- [2] S. Abarbanel, A. Ditkowski, A. Yefet, Bounded error schemes for the wave equation on complex domains, J. Sci. Comput. , Vol. 26, No. 1, 2006.
- [3] P. Angot, C.-H. Bruneau, P. Fabrie, A penalization method to take into account obstacles in viscous flows, Numer. Math. 81 No. 4 (1999), pp. 497-520.
- [4] E. Arquis, J -P. Caltagirone, Sur les conditions hydrodynamiques au voisinage dune interface milieu fluide milieu poreux: application la convection naturelle, Comptes Rendus de lAcademie des Sciences Paris II 299 (1984) 14.
- [5] F. Auteri, N. Parolini, L. Quartapelle, Numerical investigation on the stability of singular driven cavity flow, J. Comput. Phys. 183(1) (2002)1-25.
- [6] F. Auteri, L. Quartapelle, L. Vigeveno, Accurate $\omega - \psi$ spectral solution of the singular driven cavity problem, J. Comp. Phys. 180(2), (2002)597-615.
- [7] J. P. Boyd, A comparison of numerical algorithms for Fourier extension of the first, second, and third kinds, J. Comput. Phys. 178(1) (2002)118-160.
- [8] J. P. Boyd, Fourier embedded domain methods: extending a function defined on an irregular region to a rectangle so that the extension is spatially periodic and C^∞ . Appl. Math. Comput. 161(2) (2005) 591-597 .
- [9] O.P. Bruno, M. Lyon, High-order unconditionally stable FC-AD solvers for general smooth domains I. Basic elements, J. Comput. Phys. 229(6) (2010) 2009-2033.
- [10] A. Bueno-Orovio, V. M. Pérez-García, F. H. Fenton, Spectral methods for partial differential equations in irregular domains: The spectral smoothed boundary method, SIAM J. Sci. Comput. Vol. 28 No.3 (2006) 886-900.

- [11] D. Calhoun, A Cartesian grid method for solving the streamfunction vorticity equations in irregular geometries, Ph.D. thesis, University of Washington, The USA, 1999.
- [12] D. Calhoun, A Cartesian grid method for solving the two-dimensional streamfunction–vorticity equations in irregular regions, *J. Comput. Phys.* 176(2) (2002).
- [13] C. Canuto, M. Y. Hussaini, A. Quarteroni, T.A. Zang, Spectral methods in fluid dynamics, Springer–Verlag, Berlin, 1987.
- [14] G. Carbou, P. Fabrie, Boundary layer for a penalization method for viscous incompressible flow, *Adv. Differ. Equat.* 8 (2003).
- [15] J. R. Chasnov, On the decay of two-dimensional homogeneous turbulence, *Phys. Fluids*, 9 (1) (1997).
- [16] H. J. H. Clercx, A spectral solver for the Navier-Stokes equations in the velocity–vorticity formulations for flows with two non-periodic directions, *J. Comput. Phys.* 137(1) (1997).
- [17] H. J. H. Clercx, C. H. Bruneau, The normal and oblique collision of a dipole with a no-slip boundary, *Comput. Fluids* 35(3) (2006) 245-279.
- [18] H. J. H. Clercx, A. H. Nielsen, D. J. Torres, E. A. Coutias, Two-dimensional turbulence in square and circular domains with no-slip walls, *Eur. J. Mech. B–Fluid.* 20 (2001) 557-576
- [19] H. J. H. Clercx, G. J. F. van Heijst, Dissipation of kinetic energy in two-dimensional bounded flows, *Phys. Rev. E* 65 (2002).
- [20] S.C.R. Dennis, L. Quartapelle, Direct solution of the vorticity–streamfunction ordinary differential equations by a Chebyshev approximation, *J. Comput. Phys.* 52(3) (1983) 448-463.
- [21] A. Ditzkowski, Bounded error finite difference schemes for initial boundary value problems on complex domains, Ph.D. thesis, Tel-Aviv University, Israel, 1997.
- [22] H. Dütsch, F. Durst, S. Becker, H. Lienhart, Low-Reynolds–number flow around an oscillating cylinder at low Keulegan–Carpenter numbers, *J. Fluid Mech.* 360 (1998), 360, 249-271.
- [23] E. A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed–boundary finite–difference methods for three-dimensional complex flow simulations, *J. Comput. Phys.* 161(1) (2000) 35-60.
- [24] C. Foias, D. D. Holm, E. S. Titi, The Navier–Stokes– α model of fluid turbulence, *Physica D* 152/153 (2001) 505-519.
- [25] M. Germano, Differential filters for the large eddy numerical simulation of turbulent flows, *Phys. Fluids* 29 (1986) 1755.

- [26] U. Ghia, K. N. Ghia and T. C. Shin, 1982, High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method, *J. Comp. Phys.* 48(3)(1982)387-411.
- [27] D. Gottlieb, S. Gottlieb, Spectral methods for discontinuous problems, in: D.F. Griffiths, G.A. Watson (Eds.), *Proceedings of the 20th Biennial Conference on Numerical Analysis*, University of Dundee, 2003, p. 65.
- [28] G.H. Keetels, U. D’Ortona, W. Kramer, H.J.H. Clercx, K. Schneider, G.J.F. van Heijst, Fourier spectral and wavelet solvers for the incompressible Navier-Stokes equations with volume-penalization: convergence of a dipole–wall collision, *J. Comput. Phys.* 227(2)(2007)919-945.
- [29] N.K.R. Kevlahan, J.-M. Ghidaglia, Computation of turbulent flow past an array of cylinders using a spectral method with Brinkman penalization, *Eur. J. Mech. B–Fluids.* 20 (2001) 333350.
- [30] D. Kim, H. Choi, Immersed boundary method for flow around an arbitrarily moving body, *J. Comput. Phys.* 212(2) (2006) 662-680.
- [31] D. Kolomenskiy, K. Schneider, A Fourier spectral method for the Navier–Stokes equations with volume penalization for moving solid obstacles. *J. Comput. Phys.* 228(16)(2009) 5687-5709.
- [32] D.V. Le, B.C. Khoo, and K.M. Lim, An implicit-forcing immersed boundary method for simulating viscous flows in irregular domains, *Comput. Methods Appl. Mech. Engrg.* 197(25-28) (2008) 2119-2130.
- [33] C. Liao, Y. Chang, C. Lin, J.M. McDonough, Simulating flows with moving rigid boundary using immersed–boundary method, *Comput. Fluids.* 39(1) (2010) 152-167.
- [34] M. N. Linnick, H. F. Fasel, A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains, *J. Comput. Phys.* 204 (1) (2005) 157-192.
- [35] K. Luo, Z. Wang, and J. Fan, K. Cen, Full-scale solution to particle–laden flows: Multidirect forcing and immersed boundary method, *Phys. Rev. E. Stat. Nonlin. Soft. Matter. Phys.* (2007), (6 Pt 2):066709, 76
- [36] C. Migeon, A. Texier and G. Pineau, Effects of lid–driven cavity shape on the flow establishment phase, *J. Fluid. Struct.* (2000) 14, 469-488.
- [37] J. Mohd-Yusof, Combined immersed boundary/B-splines method for simulations of flows in complex geometry, *Annual Research Briefs*, Center for Turbulence Research, 1997.
- [38] P. Orlandi, Vortex dipole rebound from a wall, *Phys. Fluids A* 2 (1990) 1429.
- [39] D. Rempfer, On boundary conditions for incompressible Navier–Stokes problems, *Appl. Mech. Rev.* 59 (3) 107 2006.

- [40] D. Russell, Z. Jane Wang, A Cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow J. Comput. Phys. 191(1) (2003)177-205.
- [41] F. Sabetghadam, S. Sharafatmandjoo, F. Norouzi, Fourier spectral embedded boundary solution of the Poissons and Laplace equations with Dirichlet boundary conditions, J. Comput. Phys. 228(1)(2009)55-74.
- [42] K. Schneider, Numerical simulation of the transient flow behaviour in chemical reactors using a penalisation method, Comput. Fluids 34(10) (2005) 1223-1238.
- [43] K. Schneider, M. Farge, Numerical simulation of the transient flow behaviour in tube bundles using a volume penalization method, J. Fluids. Struct., 20 (2005)555-566.
- [44] B. Sjögreen, N. A. Petersson, A Cartesian embedded boundary method for hyperbolic conservation laws, Commun. Comput. Phys. 2 (2007) 1199-1219.
- [45] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, J. Comput. Phys., 209(2) (2005) 448-476.
- [46] M. Vanella, E. Balaras, A moving-least-squares reconstruction for embedded-boundary formulations, J. Comput. Phys. 228(18) (2009) 66176628.
- [47] Z. Wang, J. Fan, and K. Cen, Immersed boundary method for the simulation of 2D viscous flow based on vorticity-velocity formulations, J. Comput. Phys. 228(5)(2008)1504-1520.